

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

2004.01

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

REC'D 04 MAY 2001
WIPO PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

2000年 9月 5日

出願番号
Application Number:

特願2000-268043

出願人
Applicant(s):

ソニー株式会社

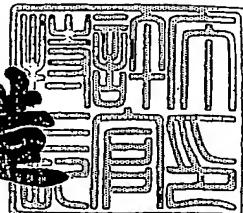
PRIORITY
DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2001年 3月 2日

特許庁長官
Commissioner,
Patent Office

及川耕三



出証番号 出証特2001-3014435

【書類名】 特許願
【整理番号】 0000554606
【提出日】 平成12年 9月 5日
【あて先】 特許庁長官殿
【国際特許分類】 H04N 5/76
【発明者】
【住所又は居所】 東京都品川区北品川 6 丁目 7 番 35 号 ソニー株式会社
内
【氏名】 加藤 元樹
【発明者】
【住所又は居所】 東京都品川区北品川 6 丁目 7 番 35 号 ソニー株式会社
内
【氏名】 浜田 俊也
【特許出願人】
【識別番号】 000002185
【氏名又は名称】 ソニー株式会社
【代表者】 出井 伸之
【代理人】
【識別番号】 100082131
【弁理士】
【氏名又は名称】 稲本 義雄
【電話番号】 03-3369-6479
【先の出願に基づく優先権主張】
【出願番号】 特願2000-183770
【出願日】 平成12年 4月 21日
【手数料の表示】
【予納台帳番号】 032089
【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708842

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置および方法、並びに記録媒体

【特許請求の範囲】

【請求項1】 入力されるAVストリームの特徴点を検出する特徴点検出手段と、

前記特徴点検出手段により検出された前記特徴点に対応する前記AVストリーム上の少なくとも1つの位置を検出する位置検出手段と、

前記位置検出手段により検出された前記位置を、前記AVストリームに対応して、記録媒体に記録する記録手段と

を含むことを特徴とする情報処理装置。

【請求項2】 前記位置は、前記AVストリームのタイムスタンプと、前記タイムスタンプの時間軸で表される

ことを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記タイムスタンプは、プレゼンテーションタイムスタンプであり、

前記タイムスタンプの時間軸は、前記プレゼンテーションタイムスタンプが置かれているところの時間軸の識別番号で表される

ことを特徴とする請求項2に記載の情報処理装置。

【請求項4】 前記位置は、前記AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表される

ことを特徴とする請求項1に記載の情報処理装置。

【請求項5】 前記位置は、前記AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される

ことを特徴とする請求項4に記載の情報処理装置。

【請求項6】 前記特徴点検出手段により検出された前記特徴点のタイプを検出するタイプ検出手段をさらに含み、

前記記録手段は、前記位置検出手段により検出された前記位置と、前記タイプ検出手段により検出された前記タイプとを対応させて記録する

ことを特徴とする請求項1に記載の情報処理装置。

【請求項7】 前記特徴点は、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項8】 入力されるAVストリームの特徴点を検出する特徴点検出手段と、

前記特徴点検出手段により検出された前記特徴点に対応する前記AVストリーム上の少なくとも1つの位置を検出する位置検出ステップと、

位置検出ステップの処理により検出された前記位置を、前記AVストリームに対応して記録媒体に記録する記録ステップと
を含むことを特徴とする情報処理方法。

【請求項9】 入力されるAVストリームの特徴点を検出する特徴点検出手段と、

前記特徴点検出手段により検出された前記特徴点に対応する前記AVストリーム上の少なくとも1つの位置を検出する位置検出ステップと、

位置検出ステップの処理により検出された前記位置を、前記AVストリームに対応して情報記録媒体に記録する記録ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項10】 AVストリームの特徴点の位置が記録されている記録媒体から、前記位置を読み取る読み取り手段と、

ユーザにより対象として指定された前記特徴点を取得する取得手段と、

前記取得手段により取得された前記特徴点に対応する前記AVストリーム上の位置を、前記読み取り手段の読み取り結果に基づいて検出する位置検出手段と、

前記位置検出手段により検出された前記AVストリーム上の位置を検索し、前記記録媒体のその位置から、前記AVストリームを再生する再生手段と

を含むことを特徴とする情報処理装置。

【請求項11】 前記AVストリームのサムネイルのリストをユーザに提示する提示手段をさらに含み、

前記取得手段は、前記提示手段により提示された前記リスト上で指定された前記サムネイルに対応する特徴点を取得する

ことを特徴とする請求項10に記載の情報処理装置。

【請求項12】 前記位置は、前記AVストリームのタイムスタンプと、前記タイムスタンプの時間軸で表される

ことを特徴とする請求項10に記載の情報処理装置。

【請求項13】 前記タイムスタンプは、プレゼンテーションタイムスタンプであり、

前記タイムスタンプの時間軸は、前記プレゼンテーションタイムスタンプが置かれているところの時間軸の識別番号で表される

ことを特徴とする請求項12に記載の情報処理装置。

【請求項14】 前記位置は、前記AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表される

ことを特徴とする請求項10に記載の情報処理装置。

【請求項15】 前記位置は、前記AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される

ことを特徴とする請求項14に記載の情報処理装置。

【請求項16】 前記特徴点は、シーンエンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含む

ことを特徴とする請求項10に記載の情報処理装置。

【請求項17】 AVストリームの特徴点の位置が記録されている記録媒体から、前記位置を読み取る読み取りステップと、

ユーザにより対象として指定された前記特徴点を取得する取得ステップと、

前記取得ステップの処理により取得された前記特徴点に対応する前記AVストリーム上の位置を、前記読み取りステップの処理での読み取り結果に基づいて検出する位置検出ステップと、

前記位置検出ステップの処理により検出された前記AVストリーム上の位置を検索し、前記記録媒体のその位置から、前記AVストリームを再生する再生ステップ

と

を含むことを特徴とする情報処理方法。

【請求項18】 AVストリームの特徴点の位置が記録されている情報記録媒体から、前記位置を読み取る読み取りステップと、

ユーザにより対象として指定された前記特徴点を取得する取得ステップと、

前記取得ステップの処理により取得された前記特徴点に対応する前記AVストリーム上の位置を、前記読み取りステップの処理での読み取り結果に基づいて検出する位置検出ステップと、

前記位置検出ステップの処理により検出された前記AVストリーム上の位置を検索し、前記情報記録媒体のその位置から、前記AVストリームを再生する再生ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は情報処理装置および方法、並びに記録媒体に関し、特に、AVストリームの所望の位置に、迅速にアクセスすることができるようとした情報処理装置および方法、並びに記録媒体に関する。

【0002】

【従来の技術】

近年、記録可能で記録再生装置から取り外し可能なディスク型媒体として、各種の光ディスクが提案されている。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV (Audio Visual) 信号を記録するメディアとしての期待が高い。

【0003】

この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、記録装置自身が、アナログ入力のオーディオビデオ信号を、MPEG-2方式で画像圧縮して作るビットストリームや、デジタルテレビジョン放送の電波か

ら直接得られるMPEG2方式のビットストリームなどがある。一般に、デジタルテレビジョン放送では、MPEG2トランSPORTストリームが使われる。トランSPORTストリームは、トランSPORTパケットが連續したストリームであり、トランSPORTパケットは、例えば、MPEG2ビデオストリームやMPEG1オーディオストリームがパケット化されたものである。1つのトランSPORTパケットのデータ長は188バイトである。デジタルテレビジョン放送で受信されるトランSPORTストリームのAVプログラムを記録装置で光ディスクにそのまま記録すれば、ビデオやオーディオの品質を全く劣化させることなく記録することが可能である。

【0004】

【発明が解決しようとする課題】

ユーザが、光ディスクに記録されているトランSPORTストリームの中から興味のあるシーン、例えば番組の頭出し点などをサーチできるようにするために、再生装置はランダムアクセス再生ができることが求められる。

【0005】

一般に、MPEG2ビデオのストリームは、0.5秒程度の間隔でIピクチャを符号化し、それ以外のピクチャはPピクチャまたはBピクチャとして符号化される。したがって、MPEG2ビデオのストリームが記録された光ディスクから、ランダムアクセスし、ビデオ再生する場合、はじめに、Iピクチャをサーチしなければならない。

【0006】

しかしながら、従来は、光ディスクに記録されているトランSPORTストリームに、ランダムアクセスし、ビデオ再生する場合に、Iピクチャの開始バイトを効率よくサーチすることが困難であった。すなわち、光ディスク上のトランSPORTストリームのランダムなバイト位置から、読み出したビデオストリームのシンタクスを解析し、Iピクチャの開始バイトをサーチしなければならず、Iピクチャのサーチに時間がかかり、ユーザからの入力に対して応答の速いランダムアクセス再生を行うことが困難であった。

【0007】

本発明は、このような状況を鑑みてなされたものであり、ユーザのランダムアクセス再生の指示に対して、記録媒体からのトランスポートストリームの読み出し位置の決定とストリームの復号開始を速やかに行えるようにするものである。

【0008】

【課題を解決するための手段】

本発明の第1の情報処理装置は、入力されるAVストリームの特徴点を検出する特徴点検出手段と、特徴点検出手段により検出された特徴点に対応するAVストリーム上の少なくとも1つの位置を検出する位置検出手段と、位置検出手段により検出された位置を、AVストリームに対応して記録媒体に記録する記録手段とを含むことを特徴とする。

【0009】

前記位置は、AVストリームのタイムスタンプと、タイムスタンプの時間軸で表されるようにすることができる。

【0010】

前記タイムスタンプは、プレゼンテーションタイムスタンプであり、タイムスタンプの時間軸は、プレゼンテーションタイムスタンプが置かれているところの時間軸の識別番号で表されるようにすることができる。

【0011】

前記位置は、AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

【0012】

前記位置は、AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

【0013】

前記特徴点検出手段により検出された特徴点のタイプを検出するタイプ検出手段をさらに含み、記録手段は、位置検出手段により検出された位置と、タイプ検出手段により検出されたタイプとを対応させて記録するようにすることができる。

【0014】

前記特徴点は、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むようにすることができる。

【0015】

本発明の第1の情報処理方法は、入力されるAVストリームの特徴点を検出する特徴点検出手段と、特徴点検出手段により検出された特徴点に対応するAVストリーム上の少なくとも1つの位置を検出する位置検出ステップと、位置検出ステップの処理により検出された位置を、AVストリームに対応して記録媒体に記録する記録ステップとを含むことを特徴とする。

【0016】

本発明の第1の記録媒体のプログラムは、入力されるAVストリームの特徴点を検出する特徴点検出手段と、特徴点検出手段により検出された特徴点に対応するAVストリーム上の少なくとも1つの位置を検出する位置検出ステップと、位置検出ステップの処理により検出された位置を、AVストリームに対応して情報記録媒体に記録する記録ステップとを含むことを特徴とする。

【0017】

本発明の第2の情報処理装置は、AVストリームの特徴点の位置が記録されている記録媒体から、位置を読み取る読み取り手段と、ユーザにより対象として指定された特徴点を取得する取得手段と、取得手段により取得された特徴点に対応するAVストリーム上の位置を、読み取り手段の読み取り結果に基づいて検出する位置検出手段と、位置検出手段により検出されたAVストリーム上の位置を検索し、記録媒体のその位置から、AVストリームを再生する再生手段とを含むことを特徴とする。

【0018】

前記AVストリームのサムネイルのリストをユーザに提示する提示手段をさらに含み、取得手段は、提示手段により提示された前記リスト上で指定されたサムネイルに対応する特徴点を取得するようにすることができる。

【0019】

前記位置は、AVストリームのタイムスタンプと、タイムスタンプの時間軸で表

されるようにすることができる。

【0020】

前記タイムスタンプは、プレゼンテーションタイムスタンプであり、タイムスタンプの時間軸は、プレゼンテーションタイムスタンプが置かれているところの時間軸の識別番号で表されるようにすることができる。

【0021】

前記位置は、AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

【0022】

前記位置は、AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

【0023】

前記特徴点は、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むようにすることができる。

【0024】

本発明の第2の情報処理方法は、AVストリームの特徴点の位置が記録されている記録媒体から、位置を読み取る読み取りステップと、ユーザにより対象として指定された特徴点を取得する取得ステップと、取得ステップの処理により取得された特徴点に対応するAVストリーム上の位置を、読み取りステップの処理での読み取り結果に基づいて検出する位置検出ステップと、位置検出ステップの処理により検出されたAVストリーム上の位置を検索し、記録媒体のその位置から、AVストリームを再生する再生ステップとを含むことを特徴とする。

【0025】

本発明の第2の記録媒体のプログラムは、AVストリームの特徴点の位置が記録されている情報記録媒体から、位置を読み取る読み取りステップと、ユーザにより対象として指定された特徴点を取得する取得ステップと、取得ステップの処理により取得された特徴点に対応するAVストリーム上の位置を、読み取りステップの処理での読み取り結果に基づいて検出する位置検出ステップと、位置検出ステ

ップの処理により検出されたAVストリーム上の位置を検索し、情報記録媒体のその位置から、AVストリームを再生する再生ステップとを含むことを特徴とする。

【0026】

本発明の第1の情報処理装置および方法、並びに記録媒体のプログラムにおいては、検出された特徴点に対応するAVストリーム上の位置がAVストリームに対応して記録される。

【0027】

本発明の第2の情報処理装置および方法、並びに記録媒体のプログラムにおいては、記録媒体からAVストリーム特徴点の位置が読み取られ、取得された特徴点に対応する位置が読み取られた結果に基づいて検出され、検出されたAVストリーム上の位置が検索され、再生される。

【0028】

【発明の実施の形態】

以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う記録部2の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0029】

端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、解析部14とAVエンコーダ15に出力される。解析部14は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

【0030】

AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

【0031】

符号化ビデオストリームは、例えば、MPEG (Moving Picture Expert Group) 2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG 1方式により符号化されたオーディオストリームや、ドルビーアンスルーパーAC3方式（商標）により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケッタイザ19に出力する。

【0032】

多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームに符号化する。AVストリームは、ECC（誤り訂正）符号化部20と変調部21でECC符号の付加と変調処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む（記録する）。

【0033】

デジタルインターフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0034】

入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ25とスイッチ17を介して多重化ストリーム解析部18と、ソースパケッタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナ

ログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0035】

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、スイッチ25を介してデマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0036】

デマルチプレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

【0037】

一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0038】

本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力さ

れるユーザからの指示情報である。

【0039】

解析部14から供給される動画像の特徴情報は、AVエンコーダ15がビデオ信号を符号化する場合において、解析部14により生成されるものである。解析部14は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像（クリップマーク）に関係する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点やCMコマーシャルのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネールも含まれる。さらにオーディオ信号のステレオとモノラルの切り換え点や、無音区間などの情報も含まれる。

【0040】

これらの画像の指示情報は、制御部23を介して、マルチプレクサ16へ入力される。マルチプレクサ16は、制御部23からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャをAVストリーム上で特定するための情報を制御部23に返す。具体的には、この情報は、ピクチャのPTS（プレゼンテーションタイムスタンプ）またはその符号化ピクチャのAVストリーム上でのアドレス情報である。制御部23は、特徴的な画像の種類とその符号化ピクチャをAVストリーム上で特定するための情報を関連付けて記憶する。

【0041】

多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、解析部18により生成される。例えば、AVストリーム内のIピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報などが含まれる。また、端子13から入力されるトランスポートストリームをトランスペアレントに記録する場合、多重化ストリーム解析部18は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

【0042】

端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【0043】

制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

【0044】

上述したアプリケーションデータベース情報についての詳細は後述する。

【0045】

このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生部3により再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

【0046】

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部28に指

示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

【0047】

ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0048】

また、ユーザインターフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0049】

また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時(例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネール画像リストをユーザーインターフェースに表示して、ユーザが、その中からある画像を選択することにより行われる)、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを読み出し部28へ指示する。すなわち、ユ

ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読み出し部28へ指示する。読み出し部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデパケッタイザ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点のピクチャのアドレスで示されるAVデータが再生される。

【0050】

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0051】

読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【0052】

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

【0053】

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不

必要なストリーム部分を消去するように、書き込み部22に指示する。

【0054】

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0055】

まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタイザ31を経て、デマルチプレクサ26に出力される。

【0056】

制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

【0057】

次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、AVデコーダ27に入力されるデータとマルチプレクサ16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力さ

れる。

【0058】

マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0059】

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの2つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlayListの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【0060】

1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリーポイント（Iピクチャ）は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【0061】

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザーが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイ

ン点 (IN) とアウト点 (OUT) の対で表される。従って、PlayListは、複数のPlayItemが集まることにより構成される。

【0062】

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【0063】

Virtual PlayListは、Clipのデータを共有していない。従って、Virtual PlayListが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【0064】

次に、Real PlayListの編集について説明する。図4 (A) は、Real PlayListのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

【0065】

図4 (B) は、Real PlayListのディバイド(divide:分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

【0066】

図4 (C) は、Real PlayListのコンバイン(combine:結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更され

る (Clip自体が1つにされる) ことはない。

【0067】

図5 (A) は、Real Playlist全体のデリート(delete:削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

【0068】

図5 (B) は、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【0069】

図5 (C) は、Real Playlistのミニマイズ(Minimize:最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlist にとって不必要的Clipの、対応するストリーム部分は削除される。

【0070】

上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

【0071】

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか?」といったメッセージなどを表示させることにより、確認(警告)を促した後に、ユーザの指示により削除の処理を実行、または、キャンセルする。または、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

【0072】

次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

【0073】

図6 (A) に示したように、2つのReal PlayList1, 2と、それぞれのReal PlayListに対応するClip1, 2が存在している場合に、ユーザがReal PlayList 1内の所定の区間 (In1 乃至 Out1までの区間: PlayItem1) を再生区間として指示し、続けて再生する区間として、Real PlayList2内の所定の区間 (In2 乃至 Out2までの区間: PlayItem2) を再生区間として指示したとき、図6 (B) に示すように、PlayItem1とPlayItem2から構成される1つのVirtual PlayList が作成される。

【0074】

次に、Virtual PlayList の再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加 append)、Virtual PlayListの中のPlayItem の削除などがある。また、Virtual PlayListそのものを削除することもできる。

【0075】

図7は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブバスとして登録することである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインバスのAVストリームに、付加的なオーディオストリームが、サブバスとして付加される。

【0076】

Real PlayListとVirtual PlayListで共通の操作として、図8に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の

中のPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList（図20などを参照して後述する）によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【0077】

次に、マーク（Mark）について説明する。マークは、図9に示されるように、ClipおよびPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark（クリップマーク）と呼ばれる。ClipMarkは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。PlayListを再生する時、そのPlayListが参照するClipのマークを参照して、使用する事ができる。

【0078】

PlayListに付加されるマークは、PlayListMark（プレイリストマーク）と呼ばれる。PlayListMarkは、主にユーザによってセットされる、例えば、ブックマークやリリューム点などである。ClipまたはPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0079】

ClipMarkの別のフォーマットとして、ClipMarkが参照するピクチャをAVストリームの中でのアドレスベースで指定するようにしても良い。Clipにマークをセットすることは、マーク点のピクチャを示すアドレスベースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマーク点のピクチャを示すアドレスベースの情報を除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0080】

次にサムネイルについて説明する。サムネイルは、Volume、PlayList、および

Clipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

【0081】

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【0082】

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、Playlistを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パソコンコンピュータやデジタルスチルカメラから取り込こまれた画像でもよい。

【0083】

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようになる必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnail）と称する。従って、マークサムネイルの元となる画像は、外部から取り込ん

だ画像よりも、マーク点の画像を抜き出したものが主となる。

【0084】

図11は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかるても、問題にはならない。

【0085】

また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【0086】

図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、およびClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

【0087】

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

【0088】

EP_mapは、エントリーポイント(EP)データのリストであり、それはエレメンタリーストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリーポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ(PTS)と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【0089】

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【0090】

TU_mapは、デジタルインターフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット(TU)データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

【0091】

本実施の形態では、セルフエンコードのストリームフォーマット(SESF)を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号(例えばDV)をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【0092】

SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリ

ームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【0093】

デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0094】

あるいは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0095】

例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【0096】

デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

【0097】

または、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時

にTU_mapが作成されてディスクに記録される。

【0098】

次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR (Digital Video Recording) と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、"DVR"ディレクトリを含むrootディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、および"DATA"ディレクトリを含む"DVR"ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【0099】

"DVR"ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。"DVR"ディレクトリは、4個のディレクトリを含む。"PLAYLIST"ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

【0100】

"CLIPINF"ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。"M2TS"ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。"DATA"ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【0101】

"DVR"ディレクトリは、次に示すファイルをストアする。"info.dvr"ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならぬ。ファイル名は、info.dvrに固定されるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのメニューサムネイルがなければならぬ。ファイル名は、me

mu.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0102】

“mark.thmb”ファイルは、マーク・サムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのマーク・サムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0103】

“PLAYLIST”ディレクトリは、2種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。“xxxxx.rpls”ファイルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、“xxxxx.rpls”である。ここで、“xxxxx”は、5個の0乃至9までの数字である。ファイル拡張子は、“rpls”でなければならぬとする。

【0104】

“yyyyy.vpls”ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、“yyyyy.vpls”である。ここで、“yyyyy”は、5個の0乃至9までの数字である。ファイル拡張子は、“vpls”でなければならぬとする。

【0105】

“CLIPINF”ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。“zzzzz.clpi”ファイルは、1つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、“zzzzz.clpi”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“clpi”でなければならぬとする。

【0106】

“M2TS”ディレクトリは、AVストリームのファイルをストアする。“zzzzz.m2ts”ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは

、 Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、"z zzzz.m2ts"であり、"zzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

【0107】

"DATA" ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【0108】

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、"info.dvr" ディレクトリについて説明する。図15は、"info.dvr" ディレクトリのシンタクスを示す図である。"info.dvr" ディレクトリは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakersPrivateData()である。

【0109】

図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0110】

MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにも良い。

【0111】

DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVR Volume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646

に従って、"0045"と符号化される。

【0112】

`length`は、この`length`フィールドの直後から`DVRVolume()`の最後までの`DVRVolume()`のバイト数を示す32ビットの符号なし整数で表される。

【0113】

`ResumeVolume()`は、ボリュームの中で最後に再生したReal PlayListまたはVirtual PlayListのファイル名を記憶している。ただし、Real PlayListまたはVirtual PlayListの再生をユーザが中断した時の再生位置は、`PlayListMark()`において定義される`resume-mark`にストアされる（図42、図43）。

【0114】

図17は、`ResumeVolume()`のシンタクスを示す図である。図17に示した`ResumeVolume()`のシンタクスを説明するに、`valid_flag`は、この1ビットのフラグが1にセットされている場合、`resume_PlayList_name`フィールドが有効であることを示し、このフラグが0にセットされている場合、`resume_PlayList_name`フィールドが無効であることを示す。

【0115】

`resume_PlayList_name`の10バイトのフィールドは、リジュームされるべきReal PlayListまたはVirtual PlayListのファイル名を示す。

【0116】

図16に示した`DVRVolume()`のシンタクスのなかの、`UIAppInfoVolume`は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、`UIAppInfoVolume`のシンタクスを示す図であり、そのセマンティクスを説明するに、`character_set`の8ビットのフィールドは、`Volume_name`フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0117】

`name_length`の8ビットフィールドは、`Volume_name`フィールドの中に示されるボリューム名のバイト長を示す。`Volume_name`のフィールドは、ボリュームの名称を示す。このフィールドの中の左から`name_length`数のバイト数が、有効なキ

キャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0118】

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事（再生される事）が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【0119】

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

【0120】

PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【0121】

次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)

のファイル名をストアする。ボリュームに記録されているすべてのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

【0122】

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0123】

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、Playlist_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。Playlist_file_nameの10バイトの数字は、PlayListのファイル名を示す。

【0124】

図21は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlaylist(後述)を含ませた構成とされている。このように、UIAppinfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

【0125】

図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでも良い。

【0126】

所定のメーカーが、プライベートデータを挿入したい時に、すでに他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

【0127】

図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

【0128】

mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカープライベートデータのエントリー数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

【0129】

mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定され

る。

【0130】

`maker_model_code`は、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。`maker_model_code`に符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。`start_mpd_block_number`は、そのメーカプライベートデータが開始される`mpd_block`の番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、`mpd_block`の先頭にアラインされなければならない。`start_mpd_block_number`は、`mpd_block`のfor-loopの中の変数jに対応する。

【0131】

`mpd_length`は、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。`mpd_block`は、メーカプライベートデータがストアされる領域である。`MakersPrivateData()`の中のすべての`mpd_block`は、同じサイズでなければならない。

【0132】

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、`xxxxxx.rpls`と`yyyyy.vpls`について説明する。図23は、`xxxxx.rpls` (Real PlayList)、または、`yyyyy.vpls` (Virtual PlayList) のシンタクスを示す図である。`xxxxx.rpls`と`yyyyy.vpls`は、同一のシンタクス構成をもつ。`xxxxx.rpls`と`yyyyy.vpls`は、それぞれ、3個のオブジェクトから構成され、それらは、`PlayList()`、`PlayListMark()`、および`MakersPrivateData()`である。

【0133】

`PlayListMark_Start_address`は、`PlayList`ファイルの先頭のバイトからの相対バイト数を単位として、`PlayListMark()`の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0134】

`MakersPrivateData_Start_address`は、`PlayList`ファイルの先頭のバイトからの相対バイト数を単位として、`MakersPrivateData()`の先頭アドレスを示す。相

対バイト数はゼロからカウントされる。

【0135】

`padding_word` (パディングワード) は、`PlayList`ファイルのシンタクスにしたがって挿入され、`N1`と`N2`は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにも良い。

【0136】

ここで、既に、簡便に説明したが、`PlayList`についてさらに説明する。ディスク内にあるすべての`Real PlayList`によって、`Bridge-Clip` (後述) を除くすべての`Clip`の中の再生区間が参照されていなければならない。かつ、2つ以上の`Real PlayList`が、それらの`PlayItem`で示される再生区間を同一の`Clip`の中でオーバーラップさせてはならない。

【0137】

図24を参照してさらに説明するに、図24 (A) に示したように、全ての`Clip`は、対応する`Real PlayList`が存在する。この規則は、図24 (B) に示したように、編集作業が行われた後においても守られる。従って、全ての`Clip`は、どちらかしらの`Real PlayList`を参照することにより、必ず視聴することが可能である。

【0138】

図24 (C) に示したように、`Virtual PlayList`の再生区間は、`Real PlayList`の再生区間または`Bridge-Clip`の再生区間の中に含まれていなければならない。どの`Virtual PlayList`にも参照されない`Bridge-Clip`がディスクの中に存在してはならない。

【0139】

`Real PlayList`は、`PlayItem`のリストを含むが、`SubPlayItem`を含んではならない。`Virtual PlayList`は、`PlayItem`のリストを含み、`PlayList()`の中に示される`CPI_type`が`EP_map type`であり、かつ`PlayList_type`が0 (ビデオとオーディオを含む`PlayList`) である場合、`Virtual PlayList`は、ひとつの`SubPlayItem`を含む事ができる。本実施の形態における`PlayList()`では、`SubPlayItem`はオーディオのアフレコの目的にだけに使用される、そして、1つの`Virtual PlayList`が持つ`Sub`

bPlayItemの数は、0または1でなければならない。

【0140】

次に、PlayListについて説明する。図25は、PlayListのシンタクスを示す図である。図25に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

【0141】

CPI_typeは、1ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

【0142】

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0または1である。付加的なオーディオストリームのパス(オーディオストリームパス)は、サブパスの一種である。

【0143】

次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方

法は、図19に示したテーブルに準拠する値に対応する。

【0144】

`name_length`は、8ビットフィールドであり、`PlayList_name`フィールドの中に示されるPlayList名のバイト長を示す。`PlayList_name`のフィールドは、`PlayList`の名称を示す。このフィールドの中の左から`name_length`数のバイト数が、有効なキャラクター文字であり、それは`PlayList`の名称を示す。`PlayList_name`フィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0145】

`record_time_and_date`は、`PlayList`が記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、14個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【0146】

`duration`は、`PlayList`の総再生時間を時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【0147】

`valid_period`は、`PlayList`が有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎた`PlayList`を自動消去する、といったように用いられる。例えば、2001/05/07は、"0x20010507"と符号化される。

【0148】

`maker_id`は、その`PlayList`を最後に更新したDVRプレーヤ(記録再生装置1)の製造者を示す16ビットの符号なし整数である。`maker_id`に符号化される値は、DVRフォーマットのライセンサによって割り当てられる。`maker_code`は、その`PlayList`を最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし

整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【0149】

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【0150】

write_protect_flagは、図28（A）にテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

【0151】

write_protect_flagが0にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていても良い。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、または、そのReal PlayListを"Minimize"する。

【0152】

is_played_flagは、図28（B）に示すように、フラグが1にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

【0153】

archiveは、図28（C）に示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail

l_index のフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

【0154】

次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【0155】

PlayListが2つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図29に示す。

【0156】

図30は、PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time (図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time (図の中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInf

o()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

【0157】

現在のPlayItemのIN_time (図の中でIN_time2と示されているもの) は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time (図の中でOUT_time2と示されているもの) は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【0158】

図3 1に示すように、PlayList()のCPI_typeがTU_map_typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【0159】

PlayItemのシンタクスは、図3 2に示すようになる。図3 2に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【0160】

STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map_typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図3 3に示すように、PlayList()において定義されるCPI_typeによって異なる。

【0161】

OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図3 4に示すように、PlayList()において

定義されるCPI_typeによって異なる。

【0162】

`Connection_Condition`は、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36は、図35に示した`Connection_Condition`の各状態について説明する図である。

【0163】

次に、`BridgeSequenceInfo`について、図37を参照して説明する。`BridgeSequenceInfo()`は、現在のPlayItemの付属情報であり、次に示す情報を持つ。`Bridge-Clip AV stream`ファイルとそれに対応する`Clip Information file`（図45）を指定する`Bridge_Clip_Information_file_name`を含む。

【0164】

また、先行するPlayItemが参照する`Clip AV stream`上のソースパケットのアドレスであり、このソースパケットに続いて`Bridge-Clip AV stream`ファイルの最初のソースパケットが接続される。このアドレスは、`RSPN_exit_from_previous_Clip`と称される。さらに現在のPlayItemが参照する`Clip AV stream`上のソースパケットのアドレスであり、このソースパケットの前に`Bridge-Clip AV stream`ファイルの最後のソースパケットが接続される。このアドレスは、`RSPN_enter_to_current_Clip`と称される。

【0165】

図37において、`RSPN_arrival_time_discontinuity`は、the `Bridge-Clip AV stream`ファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、`ClipInfo()`（図46）の中において定義される。

【0166】

図38は、`BridgeSequenceinfo`のシンタクスを示す図である。図38に示した`BridgeSequenceinfo`のシンタクスを説明するに、`Bridge_Clip_Information_file_name`のフィールドは、`Bridge-Clip AV stream`ファイルに対応する`Clip Information file`のファイル名を示す。この`Clip Information file`の`ClipInfo()`において定義される`Clip_stream_type`は、'Bridge-Clip AV stream'を示していなければ

ばならない。

【0167】

RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0168】

RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0169】

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合だけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

【0170】

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（シス

テムタイムベースの不連続点)を含んではならない。`sub path`に使われる`Clip`のオーディオサンプルのクロックは、`main path`のオーディオサンプルのクロックにロックされている。

【0171】

図40は、`SubPlayItem`のシンタクスを示す図である。図40に示した`SubPlayItem`のシンタクスを説明するに、`Clip_Information_file_name`のフィールドは、`Clip Information file`のファイル名を示し、それは`PlayList`の中で`sub path`によって使用される。この`Clip Information file`の`ClipInfo()`において定義される`Clip_stream_type`は、`Clip AV stream`を示していなければならぬ。

【0172】

`SubPath_type`の8ビットのフィールドは、`sub path`のタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【0173】

`sync_PlayItem_id`の8ビットのフィールドは、`main path`の時間軸上で`sub path`が再生開始する時刻が含まれる`PlayItem`の`PlayItem_id`を示す。所定の`PlayItem`に対応する`PlayItem_id`の値は、`PlayList()`において定義される(図25参照)。

【0174】

`sync_start PTS of PlayItem`の32ビットのフィールドは、`main path`の時間軸上で`sub path`が再生開始する時刻を示し、`sync_PlayItem_id`で参照される`PlayItem`上のPTS(Presentaiotn Time Stamp)の上位32ビットを示す。`SubPath_IN_time`の32ビットフィールドは、`Sub path`の再生開始時刻をストアする。`SubPath_IN_time`は、`Sub Path`の中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

【0175】

`SubPath_OUT_time`の32ビットフィールドは、`Sub path`の再生終了時刻をストアする。`SubPath_OUT_time`は、次式によって算出される`Presenation_end_TS`の値の上位32ビットを示す。

`Presentation_end_TS = PTS_out + AU_duration`

ここで、`PTS_out`は、`SubPath`の最後のプレゼンテーションユニットに対応する33ビット長の`PTS`である。`AU_duration`は、`SubPath`の最後のプレゼンテーションユニットの90 kHz単位の表示期間である。

【0176】

次に、図23に示した`xxxxx.rpls`と`yyyyy.vpls`のシンタクス内の`PlayListMark()`について説明する。`PlayList`についてのマーク情報は、この`PlayListMark`にストアされる。図42は、`PlayListMark`のシンタクスを示す図である。図42に示した`PlayListMark`のシンタクスについて説明するに、`version_number`は、この`PlayListMark()`のバージョンナンバを示す4個のキャラクター文字である。`version_number`は、ISO 646に従って、"0045"と符号化されなければならない。

【0177】

`length`は、この`length`フィールドの直後から`PlayListMark()`の最後までの`PlayListMark()`のバイト数を示す32ビットの符号なし整数である。`number_of_PlayList_marks`は、`PlayListMark`の中にストアされているマークの個数を示す16ビットの符号なし整数である。`number_of_PlayList_marks`は、0であってもよい。`mark_type`は、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

【0178】

`mark_time_stamp`の32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。`mark_time_stamp`のセマンティクスは、図44に示すように、`PlayList()`において定義される`CPI_type`によって異なる。`PlayItem_id`は、マークが置かれているところの`PlayItem`を指定する8ビットのフィールドである。所定の`PlayItem`に対応する`PlayItem_id`の値は、`PlayList()`において定義される（図25参照）。

【0179】

`character_set`の8ビットのフィールドは、`mark_name`フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。`name_length`の8ビットフィールドは、`Mark_name`フィールドの

中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【0180】

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない事を示す。

【0181】

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル) は、図4 5に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakersPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の"zzzzz"が使用される。

【0182】

図4 5に示したzzzzz.clpi (Clip information fileファイル) のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0183】

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭

のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0184】

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word(パディングワード)は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N1, N2, N3, N4、およびN5は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【0185】

次に、ClipInfoについて説明する。図4-6は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル(Clip AVストリームまたはBridge-Clip AVストリームファイル)の属性情報をストアする。

【0186】

図4-6に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図4-7に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【0187】

offset_SPNの32ビットのフィールドは、AVストリーム(Clip AVストリーム

またはBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

【0188】

図48に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号(相対アドレス)が、しばしば、RSPN_xxx (xxxは変形する。例。RSPN_EP_start) の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【0189】

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数(SPN_xxx)は、次式で算出される。

$$SPN_xxx = RSPN_xxx - offset_SPN$$

図49に、SPN_xxx が、5 である場合の例を示す。

【0190】

TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVR ドライブ(書き込み部22)へまたはDVR ドライブ(読み出し部28)からのAVストリームの必要な入出力のピットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【0191】

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例え

ば、01:45:30は、"0x014530"と符号化される。

【0192】

`time_controlled_flag`のフラグは、AVストリームファイルの記録モードを示す。この`time_controlled_flag`が1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\begin{aligned} \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) - \alpha &\leq \text{size_clip}(t) \\ &\leq \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha \end{aligned}$$

ここで、`TS_average_rate`は、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/secondの単位で表したものである。

【0193】

また、上式において、`t`は、秒単位で表される時間を示し、`start_time`は、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。`size_clip(t)`は、時刻`t`におけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、`start_time`から時刻`t`までに10個のソースパケットが記録された場合、`size_clip(t)`は $10 * 192$ バイトである。 α は、`TS_average_rate`に依存する定数である。

【0194】

`time_controlled_flag`が0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【0195】

`TS_average_rate`は、`time_controlled_flag`が1にセットされている場合、この24ビットのフィールドは、上式で用いている`TS_average_rate`の値を示す。`time_controlled_flag`が0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートを`TS_recording_rate`の値にセットする。次に、ビデオストリームを

可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【0196】

RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

SPN_xxx = RSPN_xxx - offset_SPN
に基づいて算出される。

【0197】

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_servece_ID_validのフラグが1である場合、servece_IDのフィールドが有効であることを示す。

【0198】

is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration deascriotor (ISO/IEC13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【0199】

servece_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservece_IDの値を示す。country_codeの24ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクターワードは、ISO8859-1で符号化される。例えば、日本は”JPN”と表され、”0x4A 0x50 0x4E”と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値’0xFF’がセットされる。

【0200】

format_identifier、original_network_ID、transport_stream_ID、servece_ID、country_code、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

【0201】

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50は、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（ただし、後述するように、Clipの最大時間長は制限されている）。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【0202】

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所

のアドレスをストアする。図5.1を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目 ($k \geq 0$) のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【0203】

図5.2は、STC_Infoのシンタクスを示す図である。図5.2に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0204】

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map_typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI_typeがEP_map_typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【0205】

num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

【0206】

RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソ

スパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

`SPN_xxx = RSPN_xxx - offset_SPN`

により算出される。

【0207】

次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【0208】

program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【0209】

図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0210】

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map_typeを示す場合、このlengthフィールドはゼロにセットされても良い

。CPI()のCPI_typeがEP_map_typeを示す場合、number_of_programsは1以上の値でなければならない。

【0211】

number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【0212】

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【0213】

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0214】

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効な

オーディオストリームを含むトランSPORTパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0215】

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【0216】

図55は、図54に示したPrograminfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【0217】

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【0218】

図59は、図54に示したPrograminfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【0219】

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波

数を示す。

【0220】

次に、図45に示したzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_mapまたは一つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならぬ。

【0221】

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す

【0222】

次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【0223】

ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオスト

リーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランSPORTパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0224】

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランSPORTパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0225】

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランSPORTパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0226】

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランSPORTパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0227】

EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図6.8を参照）。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_

map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

【0228】

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明するに、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【0229】

number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム（ビデオまたはオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

【0230】

num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

【0231】

padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない

い。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【0232】

図72は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図72に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

【0233】

RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。または、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースパケットの相対アドレスを示す。

【0234】

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

`SPN_xxx = RSPN_xxx - offset_SPN`

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【0235】

次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケッ

トのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【0236】

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k ($k \geq 0$)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基いて算出される。

$$TU_{start_time}(k) = offset_time + k * time_unit_size$$

TU_start_time(k)は、45kHzの精度を持つ。

【0237】

図74は、TU_mapのシンタクスを示す図である。図74に示したTU_mapのシンタクスを説明するに、offset_timeの32bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなければならない。

【0238】

time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size <= 4500) にすることが良い。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリー数を示す。

【0239】

RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞ

れのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【0240】

図45に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

【0241】

図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0242】

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_Clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_Clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

【0243】

mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは

、図77に示すように、PlayList()の中のCPI_typeにより異なる。

【0244】

STC_sequence_idは、CPI()の中のCPI_typeがEP_map_typeを示す場合、この8ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map_typeを示す場合、この8ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0245】

name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0246】

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

【0247】

図78は、図75に代わるClipMarkの他のシンタクスを示す図であり、図79は、その場合における、図76に代わるmark_typeのテーブルの例を示す。reserved_for_maker_IDは、mark_typeが、0xC0から0xFFの値を示す時に、そのmark_typeを定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサーが指定する。mark_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は

後述する。`representative_picture_entry()`は、`mark_entry()`によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

【0248】

ClipMarkは、ユーザーがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI(グラフィカルユーザーインターフェース)を使用して、ClipMarkの情報をユーザーに提示する。ClipMarkの情報を視覚的に表示するためには、`mark_entry()`が示すピクチャよりもむしろ`representative_picture_entry()`が示すピクチャを示したほうが良い。

【0249】

図80に、`mark_entry()`と`representative_picture_entry()`の例を示す。例えば、あるプログラムが開始してから、しばらくした後(数秒後)、そのプログラムの番組名(タイトル)が表示されるとする。ClipMarkを作るときは、`mark_entry()`は、そのプログラムの開始ポイントに置き、`representative_picture_entry()`は、そのプログラムの番組名(タイトル)が表示されるポイントに置くようにしても良い。

【0250】

DVRプレーヤは、`representative_picture_entry`の画像をGUIに表示し、ユーザーがその画像を指定すると、DVRプレーヤは、`mark_entry`の置かれたポイントから再生を開始する。

【0251】

`mark_entry()`および`representative_picture_entry()`のシンタクスを、図81に示す。

【0252】

`mark_time_stamp`は、32ビットフィールドであり、`mark_entry()`の場合はマークが指定されたポイントを示すタイムスタンプをストアし、また`representative_picture_entry()`の場合、`mark_entry()`によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

【0253】

次に、ClipMarkを指定するために、PTSによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark_entry() と representative_picture_entry()のシンタクスの例を図82に示す。

【0254】

RSPN_ref_EP_startは、mark_entry()の場合、AVストリームの中でマーク点のピクチャをデコードするためのストリームのエントリーポイントを示すソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエントリーポイントを示すソースパケットの相対アドレスを示す。RSPN_ref_EP_startの値は、EP_mapの中にRSPN_EP_startとしてストアされていなければならず、かつ、そのRSPN_EP_startに対応するPTS_EP_startの値は、EP_mapの中で、マーク点のピクチャのPTSより過去で最も近い値でなければならない。

【0255】

offset_num_picturesは、32ビットのフィールドであり、RSPN_ref_EP_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、ゼロからカウントされる。図83の例の場合、offset_num_picturesは6となる。

【0256】

次に、ClipMarkを指定するために、アドレスベースの情報を使用する場合のmark_entry() と representative_picture_entry()のシンタクスの別の例を図84に示す。

【0257】

RSPN_mark_pointは、mark_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

【0258】

RSPN_mark_pointは、ソースパケット番号を単位とする大きさであり、AVスト

リームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset_SPNの値を初期値としてカウントされる。

【0259】

図85を用いて、ClipMarkとEP_mapの関係を説明する。この例の場合、EP_mapが、エントリーポイントのアドレスとしてI0, I1, Inを指定しており、これらのアドレスからシーケンスヘッダに続くIピクチャが開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエントリーポイントであるI1からデータを読み出し開始すれば良い。

【0260】

MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

【0261】

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail1()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、および、それぞれのPlayListを代表する画像をストアする。すべてのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

【0262】

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。すべてのPlayListおよびClipに対するすべてのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail1()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は一つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0263】

図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す4個のキャラクタ一文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0264】

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersP
rivateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_st
art_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、
最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。
相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()
の中に含まれているサムネイル画像のエントリー数を与える16ビットの符号
なし整数である。

【0265】

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlayList()、PlayListMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【0266】

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは"m

enu.thmb" 内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I -picture)をとらなければならない。

【0267】

`picture_data_size`は、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。`start_tn_block_number`は、サムネイル画像のデータが始まる`tn_block`の`tn_block`番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、`tb_block`の先頭と一致していなければならない。`tn_block`番号は、0から始まり、`tn_block`のfor-ループ中の変数kの値に関係する。

【0268】

`x_picture_length`は、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。`y_picture_length`は、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。`tn_block`は、サムネイル画像がストアされる領域である。`Thumbnail()`の中のすべての`tn_block`は、同じサイズ（固定長）であり、その大きさは`tn_block_size`によって定義される。

【0269】

図89は、サムネイル画像データがどのように`tn_block`に格納されるかを模式的に表した図である。図89のように、各サムネイル画像データは`tn_block`の先頭から始まり、1`tn_block`を超える大きさの場合は、連続する次の`tn_block`を使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応する事ができるようになる。

【0270】

次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ（図14）にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【0271】

まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2 トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2 トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

【0272】

1つのAligned unitは、32個のソースパケットから成る。DVR MPEG2 トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット (PID=0x1FFFのトランスポートパケット) を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2 トランスポートストリームに余分な情報を付加してはならない。

【0273】

図91に、DVR MPEG-2 トランスポートストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2 トランスポートストリームは、このモデルに従う。

【0274】

MPEG-2 トランスポートストリームの入力タイミングについて説明する。入力MPEG2 トランスポートストリームは、フルトランスポートストリームまたはパーシャルトランスポートストリームである。入力されるMPEG2 トランスポートストリームは、ISO/IEC13818-1 ([2])またはISO/IEC13818-9 ([5])に従っていなければならない。MPEG2 トランスポートストリームのi番目のバイトは、T-STD (ISO/IEC

13818-1で規定されるTransport stream system target decoder) 5 1とソースパケットタイザ(sourse packetizer) 5 4へ、時刻t(i)に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

【0275】

27MHz PLL 5 2は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) 5 3は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。Arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter 5 3のカウント値である。

【0276】

source packetizer 5 4は、すべてのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がT-STD 5 1とソースパケットタイザ 5 4の両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

【0277】

2つの連續して入力されるトランスポートパケットの時間間隔が、230/2700000秒 (約40秒) 以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【0278】

スムージングバッファ (smoothing buffer) 5 5は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ 5 5は、オーバーフローしてはならない。Rmaxは、スムージングバッファ 5 5が空でない時のスムージングバッファ 5 5からのソースパケットの出力ビットレートである。スムージングバッファ 5 5が空である時、スムージングバッファ 5 5からの出力ビ

ットレートはゼロである。

【0279】

次に、DVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$Rmax = TS_recording_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【0280】

入力トランSPORTストリームがSESFトランSPORTストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【0281】

入力トランSPORTストリームがSESFトランSPORTストリームの場合、スムージングバッファ55の大きさ (smoothing buffer size) はゼロである。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、スムージングバッファ55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い

【0282】

記録機（レコーダ）および再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytesである。

【0283】

次に、DVR MPEG-2トランSPORTストリームのプレーヤモデルについて説明す

る。図92は、DVR MPEG-2トランSPORTストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

【0284】

27MHz X-tal (クリスタル発振器) 61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 $+/-30\text{ ppm}$ ($27000000 +/- 810\text{ Hz}$)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。arrival_time_clock(i)は、時刻t(i)におけるarrival_time_clock counter 62のカウント値である。

【0285】

smoothing buffer 64において、Rmaxは、スムージングバッファ64がフルでない時のスムージングバッファ64へのソースパケットの入力ビットレートである。スムージングバッファ64がフルである時、スムージングバッファ64への入力ビットレートはゼロである。

【0286】

MPEG-2トランSPORTストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランSPORTパケットは、スムージングバッファ64から引き抜かれる。Rpkは、トランSPORTパケットレートの瞬時の最大値である。スムージングバッファ64は、アンダーフローしてはならない。

【0287】

DVR MPEG-2トランSPORTストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータと同一である。

【0288】

図93は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランSPORTパケットである。図93に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図9

4に示す。図94に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、またはcopy prohibitedとすることができます。図95は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

【0289】

copy_permission_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【0290】

arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

において、arrival_time_stampによって指定される値を持つ整数値である。

【0291】

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

【0292】

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートスト

リームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

【0293】

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランSPORTストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランSPORTストリームは、後述する符号化の制限に従わなければならず、かつ後述するDVR-STDに従わなければならぬ。

【0294】

本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフロウを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるよう、データが十分な大きさの連続したブロック単位でストアされるようにする。

【0295】

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギヤップを起こさせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【0296】

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は

、Bridge-Clipを使用する方法と使用しない方法がある。

【0297】

図96は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図96においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図96に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

【0298】

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図96においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【0299】

また、図96におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図96においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0300】

図97は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItem

の関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図97におけるTS1は、Clip1 (Clip AVストリーム)の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図97においてIN_time1で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図97におけるTS2は、Clip2 (Clip AVストリーム)の影を付けられたストリームデータから成る。

【0301】

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time (図97においてOUT_time2で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0302】

図96と図97において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および/またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【0303】

ビデオピットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近の

Clipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【0304】

図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0305】

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【0306】

図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0307】

ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオ

ストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【0308】

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連續でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップ-ボトムのフィールドシーケンスは連續でなければならない。3-2 プルダウンを使用するエンコードの場合は、"top_field_first" および "repeat_first_field" フラグを書き換える必要があるかもしれない、またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【0309】

オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例。MPEG1レイヤ2、AC-3、SESF LPCM、AAC）は、同じでなければならない。

【0310】

次に、MPEG-2トランSPORTストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【0311】

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあつてはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあつても良い。TS2のエレメンタリーストリームを伝送する最初のパケットは

、ビデオパケットでなければならない。接続点におけるトランSPORTストリームは、後述するDVR-STDに従わなくてはならない。

【0312】

ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【0313】

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【0314】

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【0315】

PlayItemの制限について説明するに、先行するPlayItemのOUT_time (図9.6、図9.7において示されるOUT_time1) は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (F図9.6、図9.7において示されるIN_time2) は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【0316】

Bridge-Clipを使用する場合のデータアロケーションの制限について、図1.0.2を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip

AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【0317】

RSPN_exit_from_previous_Clip以前のClip1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【0318】

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図103を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【0319】

Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【0320】

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランSPORTストリームの生成および検証の際ににおけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際ににおけるデコード処理をモ

デル化するための概念モデルでもある。

【0321】

DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, T_{Bn}, MB_n, EB_n, TB_{sys}, B_{sys}, Rx_n, Rbx_n, Rx_{sys}, D_n, D_{sys}, O_nおよびP_{n(k)}の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデックス番号である。TB_nは、エレメンタリーストリームnのトランスポートバッファである。

【0322】

MB_nは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EB_nは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TB_{sys}は、復号中のプログラムのシステム情報のための入力バッファである。B_{sys}は、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rx_nは、データがTB_nから取り除かれる伝送レートである。Rbx_nは、PESパケットペイロードがMB_nから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【0323】

Rx_{sys}は、データがTB_{sys}から取り除かれる伝送レートである。D_nは、エレメンタリーストリームnのデコーダである。D_{sys}は、復号中のプログラムのシステム情報に関するデコーダである。O_nは、ビデオストリームnのre-ordering bufferである。P_{n(k)}は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【0324】

DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB₁, TB_nまたはTB_{sys}のバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB₁, MB₁, EB₁, TB_n, B_n, TB_{sys}およびTB_{sys}のバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。

。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【0325】

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図96に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【0326】

図105は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図105においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図105においてATC1で示される）と同じでない。

【0327】

また、TS2のシステムタイムベースの時間軸（図105においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図105においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【0328】

DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn またはTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【0329】

TS1の残りのパケットは、TS_recording_rate(TS1)のピットレートでDVR-STDのTBn またはTBsysのバッファへ入力されなければならない。ここで、TS_recording

_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。従って、時刻T1からT2までの区間では、ソースパケットのarrival_time_stampは無視される。

【0330】

N1をTS1の最後のビデオパケットに続くTS1のトランSPORTパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のピットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T2 - T1 = N1 / TS_recording_rate (TS1)$$

時刻T1乃至T2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

【0331】

T2の時刻において、arrival_time_clock_counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TBn またはTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

【0332】

付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるよう、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【0333】

ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図105ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-ST

Dに入力した時刻から開始する。) とする。

【0334】

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS2startは、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、Tppは、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$\text{STC_delta} = \text{PTS1end} + \text{Tpp} - \text{PTS2start}$$

【0335】

オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である(図105に図示されている"audio overlap"を参照)。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

【0336】

DVR-STDのシステムタイムクロックについて説明するに、時刻T5において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていても良い。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値(STC1)と新しいタイムベースの値(STC2)の間で切り替える。STC2の値は、次式により算出される。

$$\text{STC2} = \text{STC1} - \text{STC_delta}$$

【0337】

バッファリングの連續性について説明する。STC11video_endは、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC1上のSTCの値である。STC22video_startは、TS2の最初のビデオパケット

の最初のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC2上のSTCの値である。STC21video_endは、STC11video_endの値をシステムタイムベースSTC2上の値に換算した値である。STC21video_endは、次式により算出される。

$$\text{STC21video_end} = \text{STC11video_end} - \text{STC_delta}$$

【0338】

DVR-STDに従うために、次の2つの条件を満たす事が要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$\text{STC22video_start} > \text{STC21video_end} + \Delta T1$$

この不等式が満たされるように、Clip1および、または、Clip2の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【0339】

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフローおよびアンダーフローさせてはならない。

【0340】

このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0341】

なお、本実施の形態は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

【0342】

次に、mark_entry()およびrepresentative_picture_entry()のシンタクスが

、図81に示されるような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

【0343】

最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータベースであるEP_Map(図70)、STC_Info(図52)、Program_Info(図54)、およびClipMark(図78)を読み出す。

【0344】

ステップS2において、制御部23は、ClipMark(図78)のrepresentative_picture_entry(図81)、またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを作成し、ユーザインターフェース入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0345】

ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応づけられているマーク点を取得する。

【0346】

ステップS4において、制御部23は、ステップS3で指定されたmark_entry(図81)のmark_time_stampのPTSと、STC_sequence_idを取得する。

【0347】

ステップS5において、制御部23は、STC_Info(図52)から、ステップS4で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0348】

ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が

開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、かつ、最も近いエントリーポイント(Iピクチャ)のあるソースパケット番号を取得する。

【0349】

ステップS7において、制御部23は、ステップS6で取得したエントリーポイントのあるソースパケット番号から、トランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0350】

ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

【0351】

以上の動作を、図107乃至109を参照してさらに説明する。

【0352】

いま、図107に示されているように、DVRトランSPORTストリームファイルは、STC_sequence_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM(コマーシャル)が挿入されているものとする。

【0353】

このとき、図70に示されるEP_Mapに対応するEP_Mapには、図108に示されるように、RSPN_EP_startで示されるA, B, Cに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)として登録される。

【0354】

また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、およびCMエンドを表すマークタイプ(図79)0x92, 0x94, 0x95の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

【0355】

mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート、およ

びCMエンドに対応して、それぞれPTS(a1),PTS(b0),PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid0とされている。

【0356】

同様に、Representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a2),PTS(b0),PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

【0357】

PTS(A) < PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランスポートストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

【0358】

次に、図110のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

【0359】

ステップS21において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、およびClipMark(図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインターフェース入出力としての端子24からCMスキップ再生を指定する。

【0360】

ステップS23において、制御部23は、マークタイプ(図79)がCM開始点(0x94)であるマーク情報のPTSと、CM終了点(0x95)であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する(図81)。

【0361】

ステップS24において、制御部23は、STC_Info(図52)からCM開始点と終了点の、STC_sequence_idに対応するSTC時間軸が開始するソースパケット番号

を取得する。

【0362】

ステップS25において、制御部23は、記録媒体100からトランSPORTストリームを読み出させ、それをAVデコーダ27に供給し、デコードを開始させる。

【0363】

ステップS26において、制御部23は、現在の表示画像がCM開始点のPTSの画像か否かを調べる。現在の表示画像がCM開始点のPTSの画像でない場合には、ステップS27に進み、制御部23は、画像の表示が継続される。その後、処理はステップS25に戻り、それ以降の処理が繰り返し実行される。

【0364】

ステップS26において、現在の表示画像がCM開始点のPTSの画像であると判定された場合、ステップS28に進み、制御部23は、AVデコーダ27を制御し、デコードおよび表示を停止させる。

【0365】

次に、ステップS29において、制御部23は、CM終了点のSTC_sequence_idに対応するSTC時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS23の処理で取得したCM終了点のPTSとから、その点のPTSより時間的に前で、かつ、最も近いエントリーポイントのあるソースパケット番号を取得する。

【0366】

ステップS30において、制御部23は、ステップS29の処理で取得したエントリーポイントのあるソースパケット番号から、トランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0367】

ステップS31において、制御部23は、AVデコーダ27を制御し、CM終了点のPTSのピクチャから表示を再開させる。

【0368】

図107乃至図109を参照して、以上の動作をさらに説明すると、CM開始点

とCM終了点は、この例の場合、STC_sequence_id=id0という共通のSTC時間軸上に存在し、そのSTC時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

【0369】

トランSPORTストリームがデコードされ、ステップS26で、表示時刻がPTS(b0)になったと判定された場合(CM開始点であると判定された場合)、AVデコーダ27により表示が停止される。そして、PTS(C)<PTS(c0)の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、PTS(c0)のピクチャから表示が再開される。

【0370】

なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

【0371】

次に、mark_entryとrepresentative_picture_entryが、図82に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

【0372】

ステップS41において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、およびClipMark(図78)の情報を取得する。

【0373】

次にステップS42において、制御部23は、ステップS41で読み出したClipMark(図78)に含まれるrepresentative_picture_entry(図82)またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0374】

ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を

指定することで行われる。

【0375】

ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures(図82)を取得する。

【0376】

ステップS45において、制御部23は、ステップS44で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0377】

ステップS46において、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_startで参照されるピクチャから(表示はしないで)、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

【0378】

以上の処理を、図113乃至図115を参照して、さらに説明する。この例においては、DVRトランスポートストリームファイルは、ソースパケット番号Aからシーンが開始しており、ソースパケット番号BからソースパケットCまでCMが挿入されている。このため、図114に示されるように、EP_mapには、RSPN_EP_startとしてのA, B, Cに対応して、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)が登録されている。

【0379】

また、図115に示されるように、シーンスタート、CMスタート、およびCMエンドのマークタイプに対応して、mark_entryとrepresentative_picture_entryが登録されている。mark_entryには、シーンスタート、CMスタート、およびCMエンドに対応して、RSPN_ref_EP_startとして、それぞれA, B, Cが登録され、offset_num_picturesとして、M1, N1, N2が登録されている。同様に、representative_picture_entryには、RSPN_ref_EP_startとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれA, B, Cが登録され、offset_num_picturesとして、M2, N1, N2がそれぞれ登録されている。

【0380】

シーンスタートに当たるピクチャから頭出して再生が指令された場合、パケット番号Aのデータから始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_picturesが、M1の値になったとき、そのピクチャから表示が開始される。

【0381】

さらに、mark_entryとrepresentative_picture_entryのシンタクスが、図82に示される構成である場合におけるCMスキップ再生の処理について、図116のフローチャートを参照して説明する。

【0382】

ステップS61において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）の情報を取得する。

【0383】

ステップS62において、ユーザがCMスキップ再生を指令すると、ステップS63において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報として、RSPN_ref_EP_STARTとoffset_num_pictures（図82）を取得する。そして、CM開始点のデータは、RSPN_ref_EP_start(1),offset_num_pictures(1)とされ、CM終了点のデータは、RSPN_ref_EP_start(2),offset_num_pictures(2)とされる。

【0384】

ステップS64において、制御部23は、RSPN_ref_EP_start(1),RSPN_ref_EP_start(2)に対応するPTSをEP_map（図70）から取得する。

【0385】

ステップS65において、制御部23は、トランスポートストリームを記録媒体100から読み出させ、AVデコーダ27に供給させる。

【0386】

ステップS66において、制御部23は、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN_re

f_EP_start(1)に対応するPTSのピクチャでない場合には、ステップS67に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS65に戻り、それ以降の処理が繰り返し実行される。

【0387】

ステップS66において、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであると判定された場合、ステップS68に進み、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset_num_pictures(1)になったとき、表示を停止させる。

【0388】

ステップS69において、制御部23は、RSPN_ref_EP_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0389】

ステップS70において、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_start(2)に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_pictures(2)になったとき、そのピクチャから表示を開始させる。

【0390】

以上の動作を、図113乃至図115を参照してさらに説明すると、まず、EP_map（図114）をもとに、パケット番号B, Cに対応する時刻PTS(B), PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がN1（図115）になったとき、表示が停止される。

【0391】

さらに、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、その値がN2（図115）になったとき、そのピクチャから表示が再開される。

【0392】

以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

【0393】

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図118のフローチャートを参照して説明する。

【0394】

ステップS81において、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、並びにClipMark(図78)の情報が取得される。

【0395】

ステップS82において、制御部23は、ClipMark(図78)のrepresentative_picture_entryまたはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0396】

ステップS83において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0397】

ステップS84において、制御部23は、ユーザから指定されたmark_entryのRSPN_mark_point(図84)を取得する。

【0398】

ステップS85において、制御部23は、マーク点のRSPN_mark_pointより前にあり、かつ、最も近いエントリーポイントのソースパケット番号を、EP_map(図70)から取得する。

【0399】

ステップS86において、制御部23は、ステップS85で取得したエントリ

一ポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0400】

ステップS87において、制御部23は、AVデコーダ27を制御し、RSPN_mark_pointで参照されるピクチャから表示を開始させる。

【0401】

以上の処理を、図119乃至図121を参照してさらに説明する。この例においては、DVRトランスポートストリームファイルが、ソースパケットAでシーンスタートし、ソースパケット番号BからCまでCMが挿入されている。このため、図120のEP_mapには、RSPN_EP_startとしてのA, B, Cに対応して、PTS_EP_startがそれぞれPTS(A), PTS(B), PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、およびCMエンドに対応して、markentryのRSPN_mark_pointとして、a1, b1, c1が、また、representative_picture_entryのRSPN_mark_pointとして、a2, b1, c1が、それぞれ登録されている。

【0402】

シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号A < a1 とすると、パケット番号Aのデータから始まるストリームからデコードが開始され、ソースパケット番号 a1 に対応するピクチャから表示が開始される。

【0403】

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

【0404】

ステップS101において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、並びにClipMark(図70)の情報を取得する。

【0405】

ステップS102において、ユーザは、CMスキップ再生を指定する。

【0406】

ステップS103において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point（図84）を取得する。そして、制御部23は、CM開始点のデータをRSPN_mark_point（1）とし、CM終了点のデータをRSPN_mark_point（2）とする。

【0407】

ステップS104において、制御部23は、記録媒体100からトランSPORTストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

【0408】

ステップS105において、制御部23は、現在の表示画像がRSPN_mark_point（1）に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point（1）に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

【0409】

ステップS105において、現在の表示画像がRSPN_mark_point（1）に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はAVデコーダ27を制御し、デコードおよび表示を停止させる。

【0410】

次に、ステップS108において、RSPN_mark_point（2）より前にあり、かつ、最も近いエントリーポイントのあるソースパケット番号がEP_map（図70）から取得される。

【0411】

ステップS109において、制御部23は、ステップS108で取得したエントリーポイントに対応するソースパケット番号からトランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0412】

ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN_mark_point（2）で参照されるピクチャから表示を再開させる。

【0413】

以上の処理を図119乃至図121の例でさらに説明すると、Clip AV streamをデコードして行き、ソースパケット番号b1（図121）に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応するピクチャになったとき、そのピクチャから表示が再開される。

【0414】

以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

【0415】

より具体的には、図125に示されるように、PlayList上において、PlayList Markとしてブックマークやリリューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

【0416】

なお、ClipMarkのシンタクスは、図78の例に替えて、図126に示すようにすることもできる。

【0417】

この例においては、RSPN_markが、図78のreserved_for_MakerID, mark_entry()、およびrepresentative_picture_entry()に替えて挿入されている。このRSPN_markの32ビットのフィールドは、AVストリームファイル上で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

【0418】

その他の構成は、図78における場合と同様である。

【0419】

ClipMarkのシンタクスは、さらに図127に示すように構成することもできる。この例においては、図126におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

【0420】

以上のように、AVストリームの付属情報として、エントリーポイントのアドレスをストアするためのEP_mapと、マーク点のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのAVストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、AVストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

【0421】

このClip Information file情報により、ユーザが、記録媒体100に記録されているAVストリームの中から興味のあるシーン、例えば番組の頭出し点などをサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体100からのAVストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

【0422】

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置1は、図128に示されるようなパーソナルコンピュータにより構成される。

【0423】

図128において、CPU (Central Processing Unit) 201は、ROM (Read Only Memory) 202に記憶されているプログラム、または記憶部208からRAM (Random Access Memory) 203にロードされたプログラムに従って各種の処理を実行する。RAM 203にはまた、CPU 201が各種の処理を実行する上に

おいて必要なデータなども適宜記憶される。

【0424】

CPU201、ROM202、およびRAM203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インターフェース205も接続されている。

【0425】

入出力インターフェース205には、キーボード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モ뎀、ターミナルアダプタなどより構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

【0426】

入出力インターフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

【0427】

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0428】

この記録媒体は、図128に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピディスクを含む）、光ディスク222（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク223（MD（Mini-Disk）を含む）、若しくは半導体メモリ224などより

なるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスクなどで構成される。

【0429】

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0430】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0431】

【発明の効果】

本発明の第1の情報処理装置および方法、並びに記録媒体のプログラムによれば、検出された特徴点に対応するAVストリーム上の位置を、AVストリームに対応して記録するようにしたので、AVストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

【0432】

本発明の第2の情報処理装置および方法、並びに記録媒体のプログラムによれば、取得された特徴点に対応するAVストリーム上の位置を、読み取り結果に基づいて検出し、検出されたAVストリーム上の位置を検索し、その位置からAVストリーム再生するようにしたので、ユーザが所望する位置からの再生を、迅速且つ確実に行うことが可能となる。

【図面の簡単な説明】

【図1】

本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】

記録再生装置1により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図3】

Real PlayListとVirtual PlayListについて説明する図である。

【図4】

Real PlayListの作成について説明する図である。

【図5】

Real PlayListの削除について説明する図である。

【図6】

アセンブル編集について説明する図である。

【図7】

Virtual PlayListにサブパスを設ける場合について説明する図である。

【図8】

PlayListの再生順序の変更について説明する図である。

【図9】

PlayList上のマークとClip上のマークについて説明する図である。

【図10】

メニュー・サムネイルについて説明する図である。

【図11】

PlayListに付加されるマークについて説明する図である。

【図12】

クリップに付加されるマークについて説明する図である。

【図13】

PlayList、Clip、サムネイルファイルの関係について説明する図である。

【図14】

ディレクトリ構造について説明する図である。

【図15】

info.dvrのシンタクスを示す図である。

【図16】

DVR volumeのシンタクスを示す図である。

【図17】

Resumevolumeのシンタクスを示す図である。

【図18】

UIAppInfovolumeのシンタクスを示す図である。

【図19】

Character set valueのテーブルを示す図である。

【図20】

TableOfPlayListのシンタクスを示す図である。

【図21】

TableOfPlayListの他のシンタクスを示す図である。

【図22】

MakersPrivateDataのシンタクスを示す図である。

【図23】

xxxxx.rplsとyyyy.vplsのシンタクスを示す図である。

【図24】

PlayListについて説明する図である。

【図25】

PlayListのシンタクスを示す図である。

【図26】

PlayList_typeのテーブルを示す図である。

【図27】

UIAppinfoPlayListのシンタクスを示す図である。

【図28】

図27に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

【図29】

PlayItemについて説明する図である。

【図30】

PlayItemについて説明する図である。

【図31】

PlayItemについて説明する図である。

【図32】

PlayItemのシンタクスを示す図である。

【図33】

IN_timeについて説明する図である。

【図34】

OUT_timeについて説明する図である。

【図35】

Connection_Conditionのテーブルを示す図である。

【図36】

Connection_Conditionについて説明する図である。

【図37】

BridgeSequenceInfoを説明する図である。

【図38】

BridgeSequenceInfoのシンタクスを示す図である。

【図39】

SubPlayItemについて説明する図である。

【図40】

SubPlayItemのシンタクスを示す図である。

【図41】

SubPath_typeのテーブルを示す図である。

【図42】

PlayListMarkのシンタクスを示す図である。

【図43】

Mark_typeのテーブルを示す図である。

【図44】

Mark_time_stampを説明する図である。

【図45】

zzzzz.clipのシンタクスを示す図である。

【図4 6】

ClipInfoのシンタクスを示す図である。

【図4 7】

Clip_stream_typeのテーブルを示す図である。

【図4 8】

offset_SPNについて説明する図である。

【図4 9】

offset_SPNについて説明する図である。

【図5 0】

STC区間について説明する図である。

【図5 1】

STC_Infoについて説明する図である。

【図5 2】

STC_Infoのシンタクスを示す図である。

【図5 3】

ProgramInfoを説明する図である。

【図5 4】

ProgramInfoのシンタクスを示す図である。

【図5 5】

VideoCodingInfoのシンタクスを示す図である。

【図5 6】

Video_formatのテーブルを示す図である。

【図5 7】

frame_rateのテーブルを示す図である。

【図5 8】

display_aspect_ratioのテーブルを示す図である。

【図5 9】

AudioCodingInfoのシンタクスを示す図である。

【図6 0】

audio_codingのテーブルを示す図である。

【図61】

audio_component_typeのテーブルを示す図である。

【図62】

sampling_frequencyのテーブルを示す図である。

【図63】

CPIについて説明する図である。

【図64】

CPIについて説明する図である。

【図65】

CPIのシンタクスを示す図である。

【図66】

CPI_typeのテーブルを示す図である。

【図67】

ビデオEP_mapについて説明する図である。

【図68】

EP_mapについて説明する図である。

【図69】

EP_mapについて説明する図である。

【図70】

EP_mapのシンタクスを示す図である。

【図71】

EP_type valuesのテーブルを示す図である。

【図72】

EP_map_for_one_stream_PIDのシンタクスを示す図である。

【図73】

TU_mapについて説明する図である。

【図74】

TU_mapのシンタクスを示す図である。

【図75】

ClipMarkのシンタクスを示す図である。

【図76】

mark_typeのテーブルを示す図である。

【図77】

mark_type_stampのテーブルを示す図である。

【図78】

ClipMarkのシンタクスの他の例を示す図である。

【図79】

Mark_typeのテーブルの他の例を示す図である。

【図80】

mark_entry()とrepresentative_picture_entry()の例を示す図である。

【図81】

mark_entry()とrepresentative_picture_entry()のシンタクスを示す図である

【図82】

mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図83】

RSPN_ref_EP_startとoffset_num_picturesの関係を説明する図である。

【図84】

mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図85】

ClipMarkとEP_mapの関係を説明する図である。

【図86】

menu.thmbとmark.thmbのシンタクスを示す図である。

【図87】

Thumbnailのシンタクスを示す図である。

【図88】

thumbnail_picture_formatのテーブルを示す図である。

【図89】

tn_blockについて説明する図である。

【図90】

DVR MPEG2のトランSPORTストリームの構造について説明する図である。

【図91】

DVR MPEG2のトランSPORTストリームのレコーダモデルを示す図である。

【図92】

DVR MPEG2のトランSPORTストリームのプレーヤモデルを示す図である。

【図93】

source packetのシンタクスを示す図である。

【図94】

TP_extra_headerのシンタクスを示す図である。

【図95】

copy permission indicatorのテーブルを示す図である。

【図96】

シームレス接続について説明する図である。

【図97】

シームレス接続について説明する図である。

【図98】

シームレス接続について説明する図である

【図99】

シームレス接続について説明する図である。

【図100】

シームレス接続について説明する図である

【図101】

オーディオのオーバーラップについて説明する図である。

【図102】

BridgeSequenceを用いたシームレス接続について説明する図である。

【図103】

BridgeSequenceを用いないシームレス接続について説明する図である。

【図104】

DVR STDモデルを示す図である。

【図105】

復号、表示のタイミングチャートを示す図である。

【図106】

図81のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図107】

図81のシンタクスの場合における再生の動作を説明する図である。

【図108】

EP_mapの例を示す図である。

【図109】

ClipMarkの例を示す図である。

【図110】

図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図111】

図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図112】

図82のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図113】

図82のシンタクスの場合における再生を説明する図である。

【図114】

EP_mapの例を示す図である。

【図115】

ClipMarkの例を示す図である。

【図116】

図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図117】

図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図118】

図84のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図119】

図84のシンタクスの場合における再生を説明する図である。

【図120】

EP_mapの例を示す図である。

【図121】

ClipMarkの例を示す図である。

【図122】

図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図123】

図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図124】

アプリケーションフォーマットを示す図である。

【図125】

PlayList上のマークとClip上のマークを説明する図である。

【図126】

ClipMarkのシンタクスの他の例を示す図である。

【図127】

ClipMarkのシンタクスのさらに他の例を示す図である。

【図128】

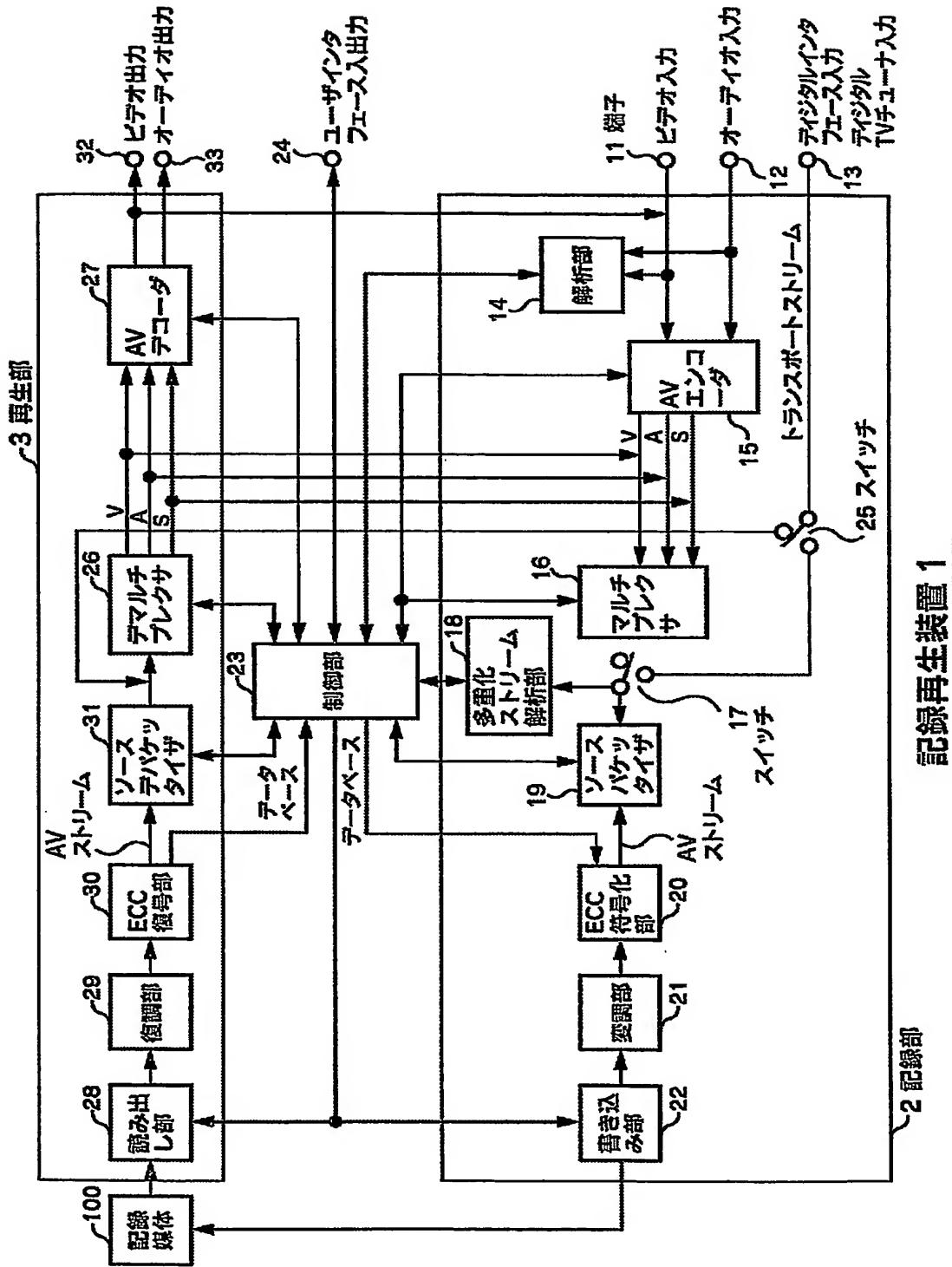
媒体を説明する図である。

【符号の説明】

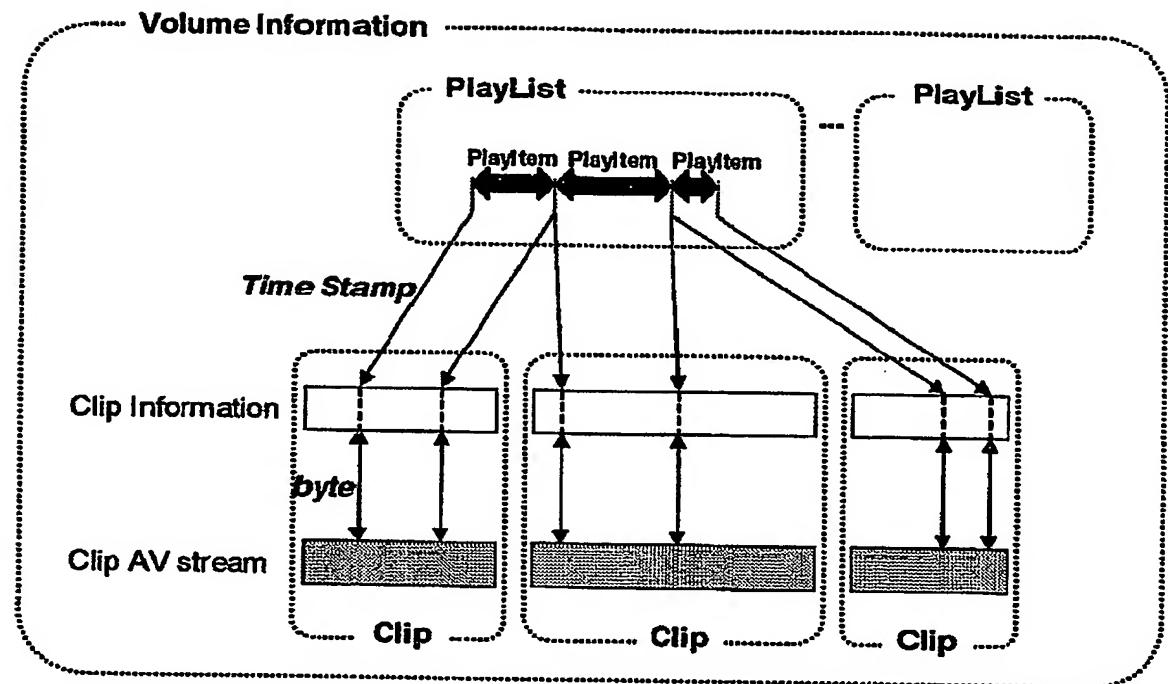
1 記録再生装置, 11乃至13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチプレクサ, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケッタイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインターフェース, 25 スイッチ, 26 デマルチプレクサ, 27 AVデコーダ, 28 読み出し部, 29 復調部, 30 ECC復号部, 31 ソースパケッタイザ, 32, 33 端子

【書類名】 図面

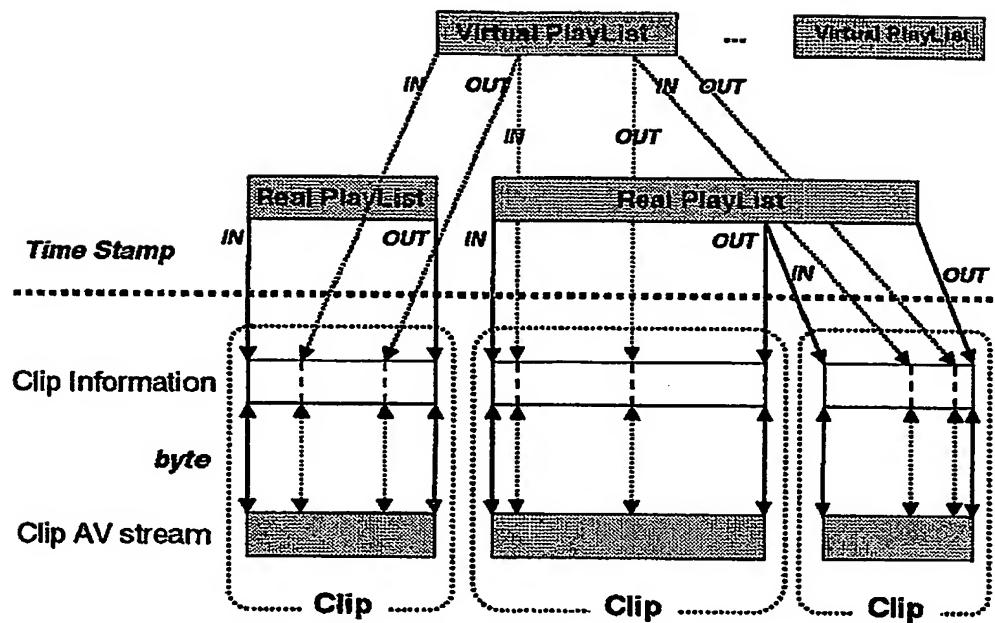
【図1】



〔図2〕

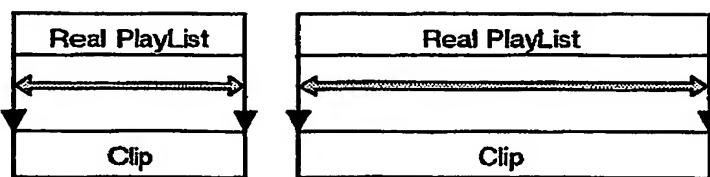


【図3】



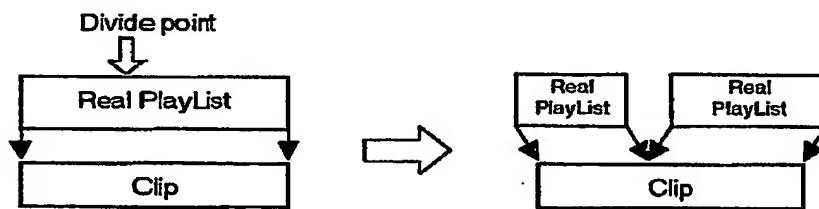
【図4】

(A)



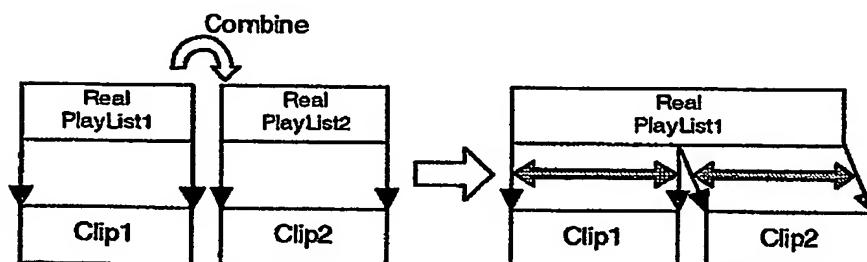
Real Playlist のクリエイトの例

(B)



Real Playlist のディバイドの例

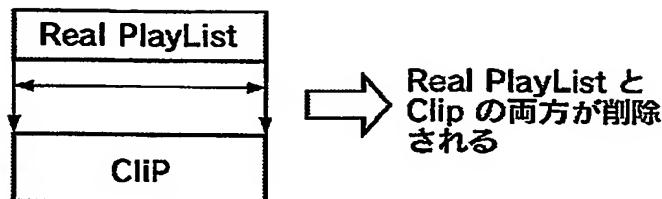
(C)



Real Playlist のコンバインの例

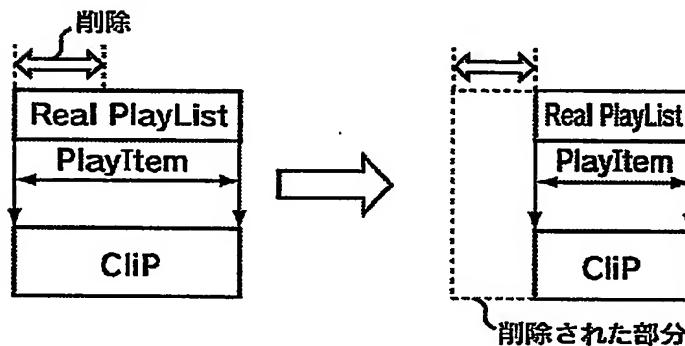
【図5】

(A)



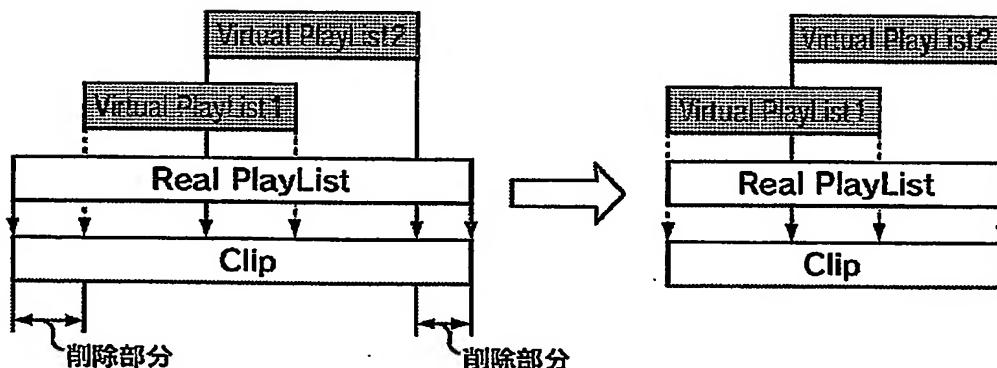
Real Playlist 全体のデリートの例

(B)



Real Playlist の部分的なデリートの例

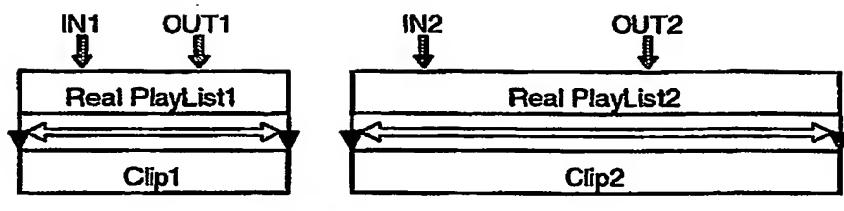
(C)



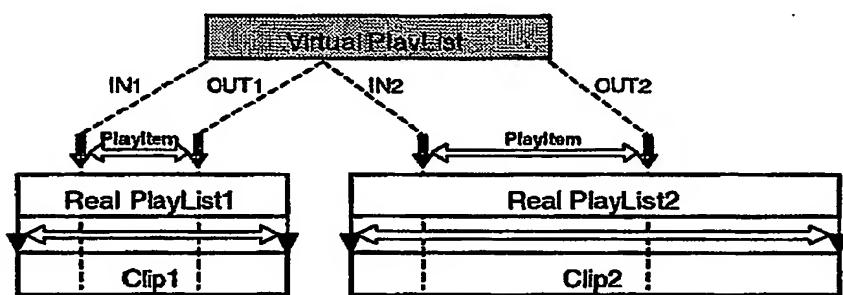
Real Playlist のミニマイズの例

【図6】

(A)

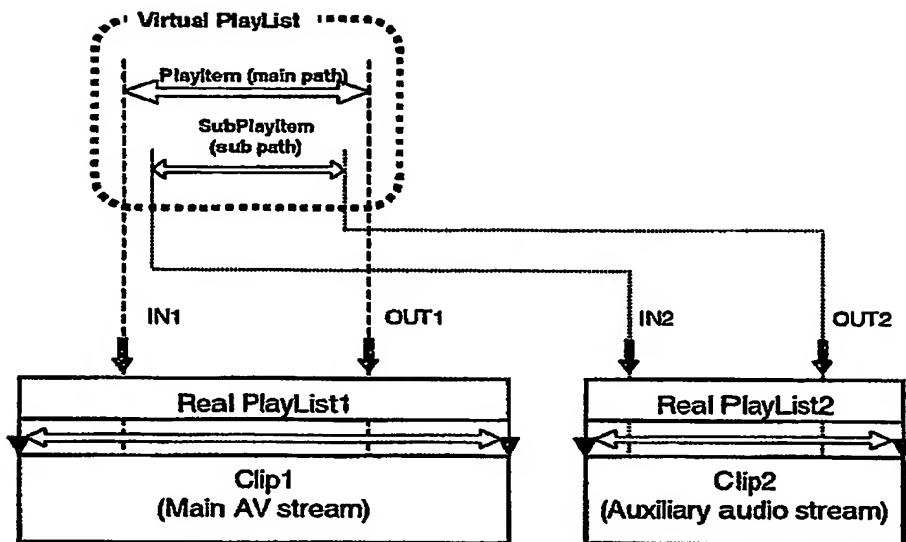


(B)



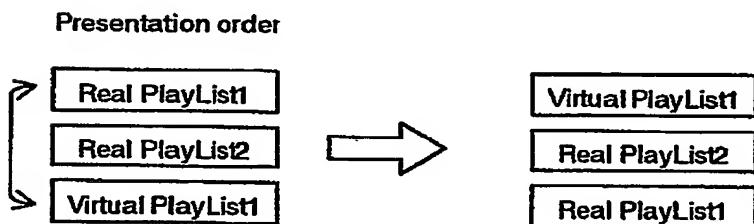
アセンブル編集の例

【図7】



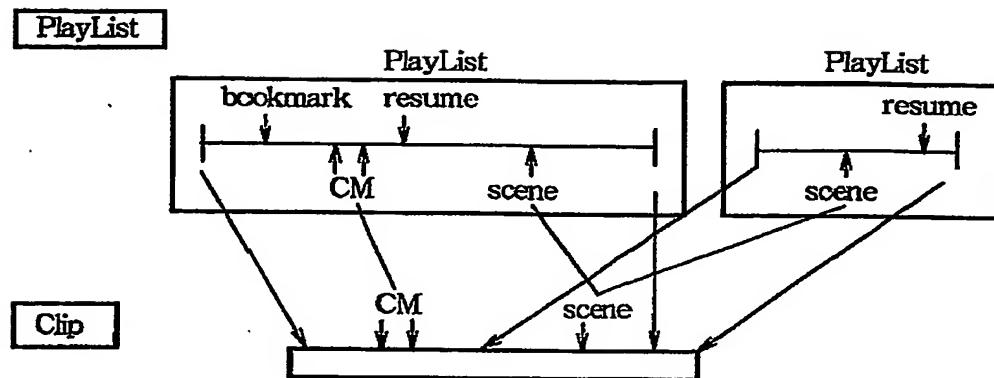
Virtual PlayListへのオーディオのアフレコの例

【図8】



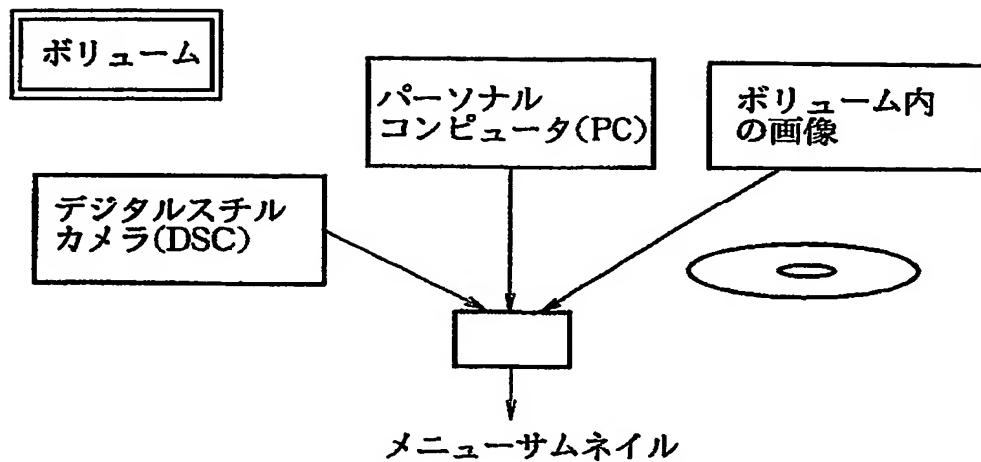
PlayListの再生順序の変更の例

【図9】

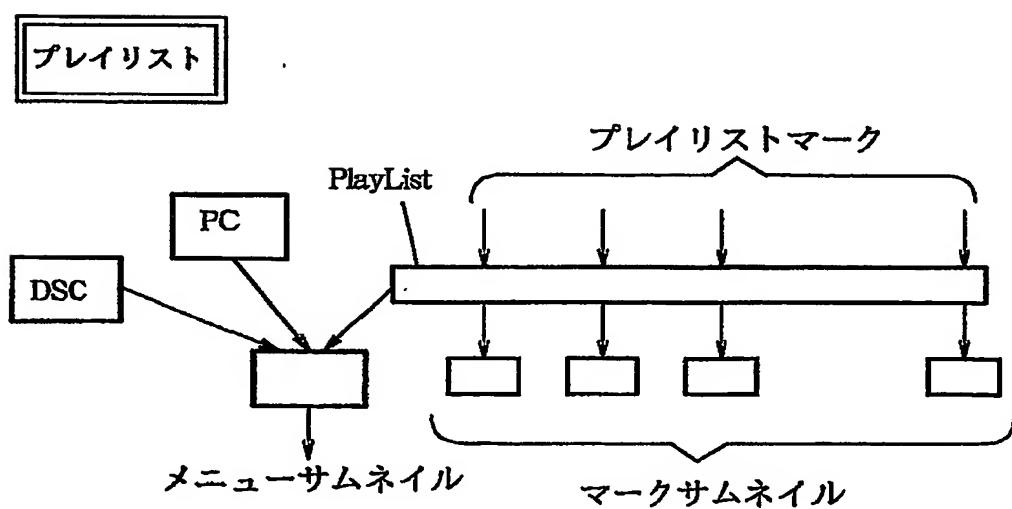


Playlist 上のマークと Clip 上のマーク

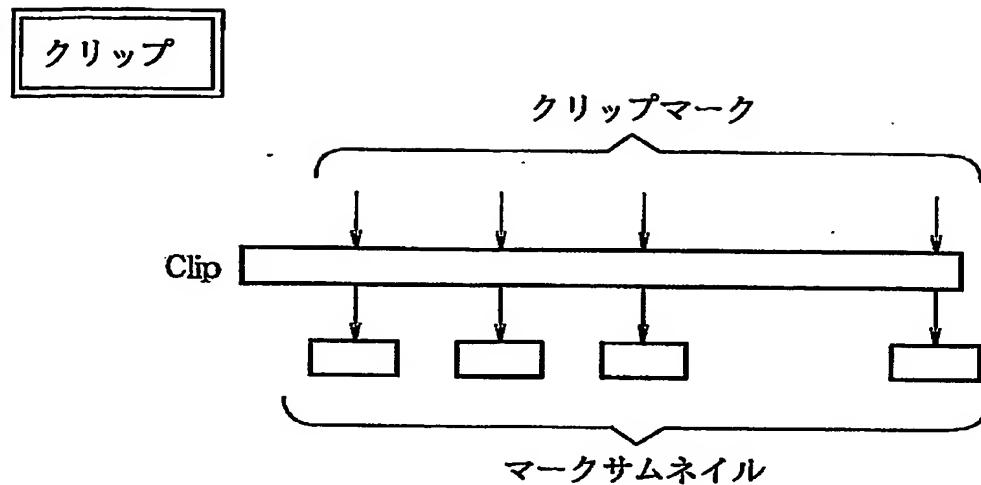
【図10】



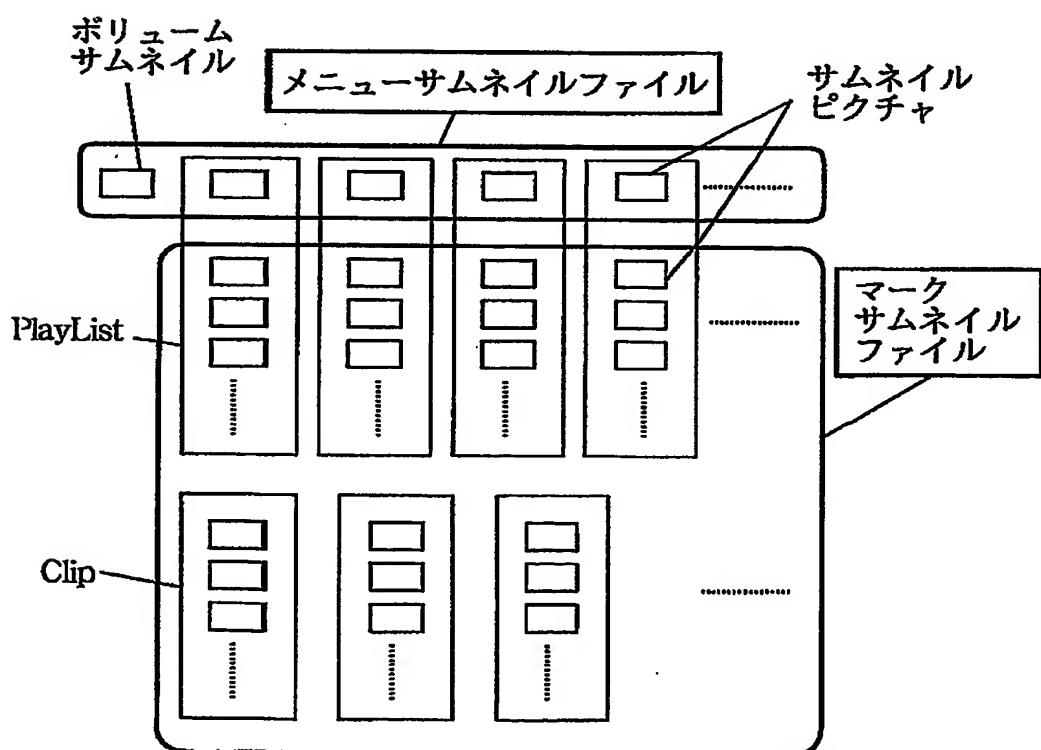
【図11】



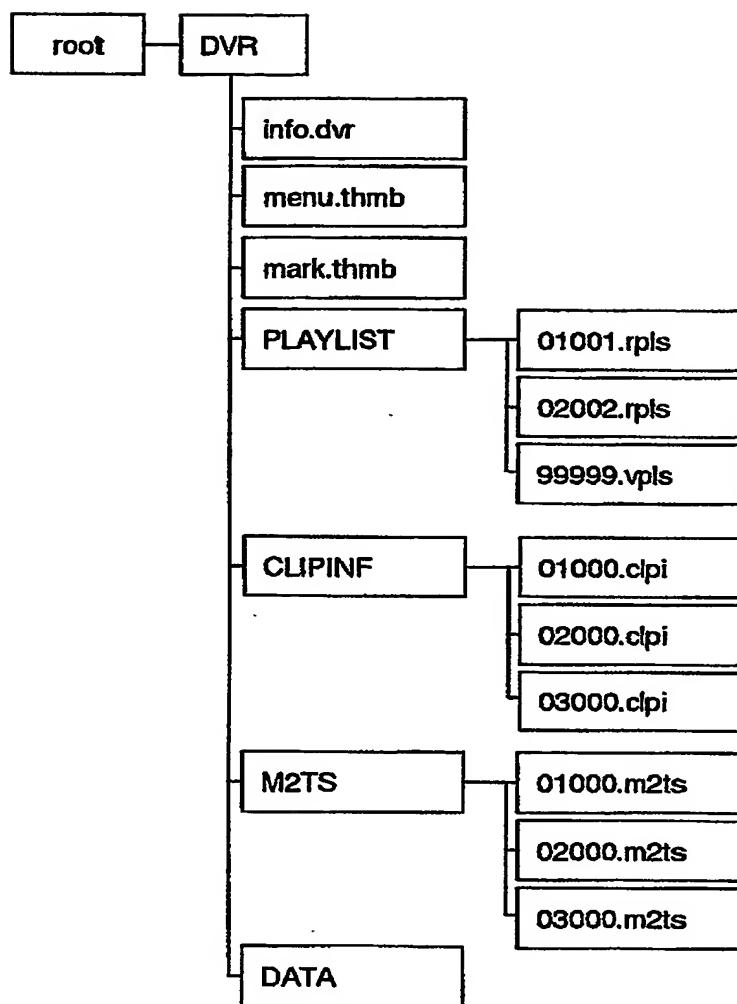
【図12】



【図13】



【図14】



【図15】

Syntax	No. of bits	Mnemonics
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

info.drv のシンタクス

【図16】

Syntax	No. of bits	Mnemonics
DVRVolume{		
version_number	8*4	bslbf
length	32	ulmsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

DVR Volume のシンタクス

【図17】

Syntax	No. of bits	Mnemonics
ResumeVolume{		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

ResumeVolume のシンタクス

【図18】

Syntax	No. of bits	No. of Mnemonics
UIAppInfoVolume {		
character_set	8	bslbf
name_length	8	ulmsbf
volume_name	8*256	bslbf
reserved	15	bslbf
volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	ulmsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoVolume のシンタクス

【図19】

Value	Character coding
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

Character set value

【図20】

Syntax	No. of bits	Mnemonics
TableOfPlayLists()		
version number	8*4	bslbf
length	32	uimsbf
number of PlayLists	16	uimsbf
for (i=0; i<number of PlayLists; i++) {		
PlayList file name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図21】

■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	ulmsbf
number_of_PlayLists	16	ulmsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

TableOfPlayLists の別シンタクス

【図22】

Syntax	No. of bits	No. of Mnemonics
MakersPrivateData()		
version_number	8*4	bslbf
length	32	ulmsbf
if(length l=0){		
mpd_blocks_start_address	32	ulmsbf
number_of_maker_entries	16	ulmsbf
mpd_block_size	16	ulmsbf
number_of_mpd_blocks	16	ulmsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	ulmsbf
maker_model_code	16	ulmsbf
start_mpd_block_number	16	ulmsbf
reserved	16	bslbf
mpd_length	32	ulmsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (i=0; i<number_of_mpd_blocks; i++){		
mpd_block	mpd_block_size*1024*8	
}		
}		

MakersPrivateData のシナリオ

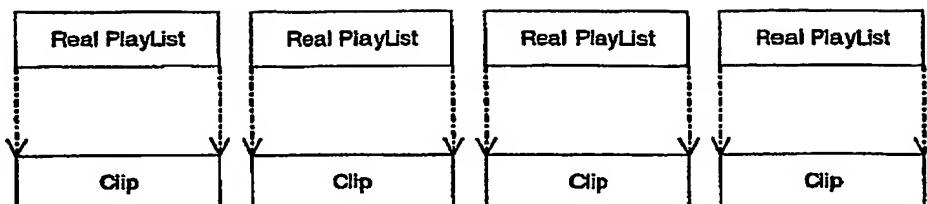
【図23】

Syntax	No. of bits	Mnemonics
xxxxx.rpls / yyyy.ypls {		
PlayListMark_Start_address	32	uimbsbf
MakersPrivateData_Start_address	32	uimbsbf
reserved	192	bslbf
PlayList()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

xxxxx.rpls と yyyy.ypls のシンタクス

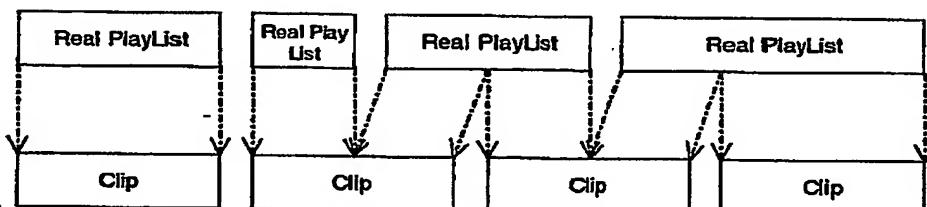
【図24】

(A)



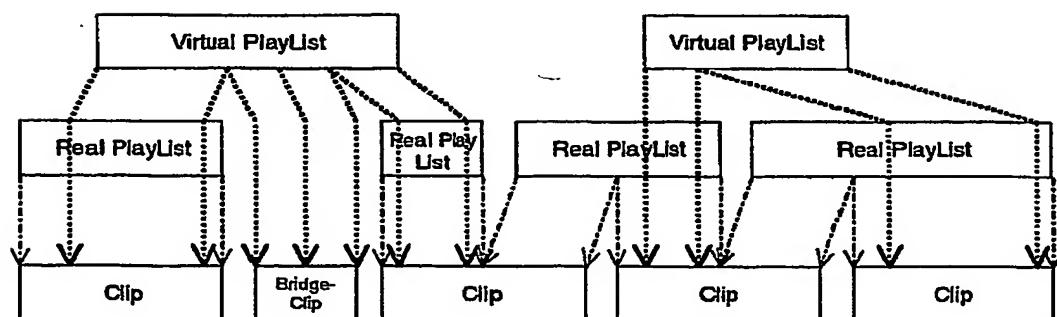
初めて AV ストリームが Clip として記録された時の Real PlayList の例

(B)



編集後の Real PlayList の例

(C)



Virtual PlayList の例

【図25】

Syntax	No. of bits	Mnemonics
PlayList()		
version number	8*4	bslbf
length	32	uimstbf
PlayList type	8	uimsbf
CPI type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number of PlayItems	16	uimstbf
if (<Virtual PlayList>){		
number of SubPlayItems	16	uimstbf
reserved	16	bslbf
}else		
for (PlayItem id=0;		
PlayItem id<number_of_PlayItems;		
PlayItem id++) {		
PlayItem()		
// main path		
}		
if (<Virtual PlayList>){		
if (CPI type==0 && PlayList type==0){		
for (i = 0; i < number of SubPlayItems; i++)		
SubPlayItem()		
// sub path		
}		
}		

Playlist のシンタクス

【図26】

PlayList_type	Meaning
0	AV記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2 - 255	reserved

PlayList_type

【図27】

Syntax		No. of bits	No. of Mnemonics
UIAppInfoPlayList2()			
character_set	8	bslbf	
name_length	8	uimsbf	
PlayList_name	8*256	bslbf	
reserved	8	bslbf	
record_time_and_date	4*14	bslbf	
reserved	8	bslbf	
duration	4*6	bslbf	
valid_period	4*8	bslbf	
maker_id	16	uimsbf	
maker_code	16	uimsbf	
reserved	11	bslbf	
playback_control_flag	1	bslbf	
write_protect_flag	1	bslbf	
is_played_flag	1	bslbf	
archive	2	bslbf	
ref_thumbnail_index	16	uimsbf	
reserved_for_future_use	256	bslbf	
}			

UIAppInfoPlayList のシンタクス

【図28】

(A)

write_protect_flag	Meaning
0b	その PlayList を自由に消去しても良い。
1b	write_protect_flag を除いてその PlayList の内容は、消去および変更されるべきではない。

write_protect_flag

(B)

is_played_flag	Meaning
0b	その PlayList は、記録されてから一度も再生されたことがない。
1b	PlayList は、記録されてから一度は再生された。

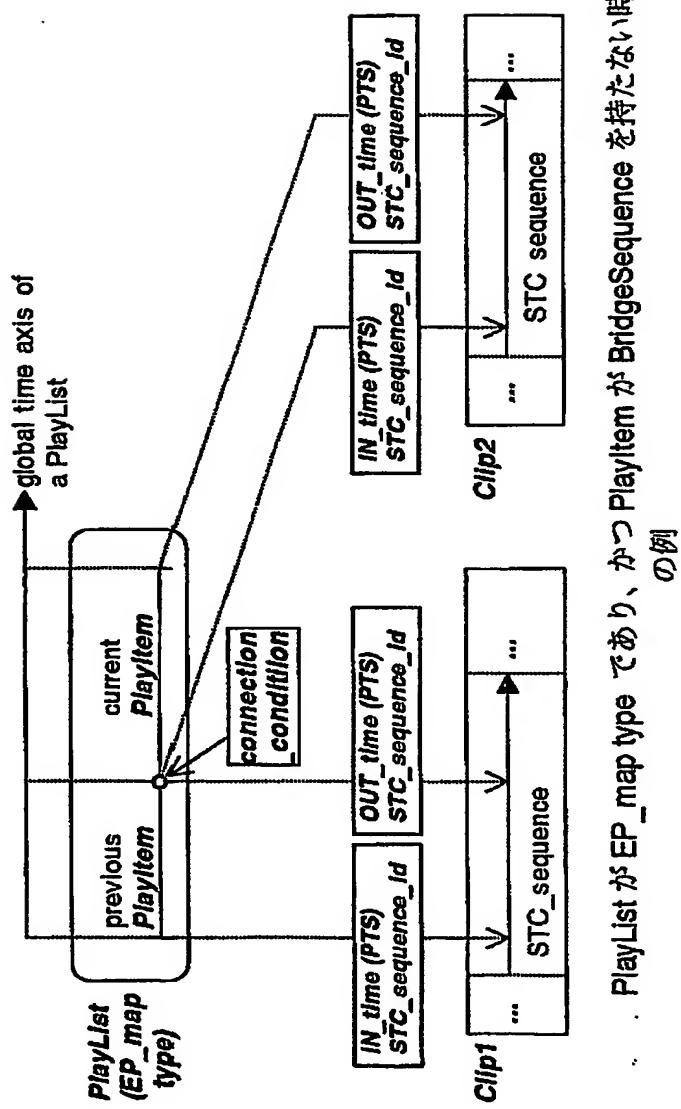
is_played_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

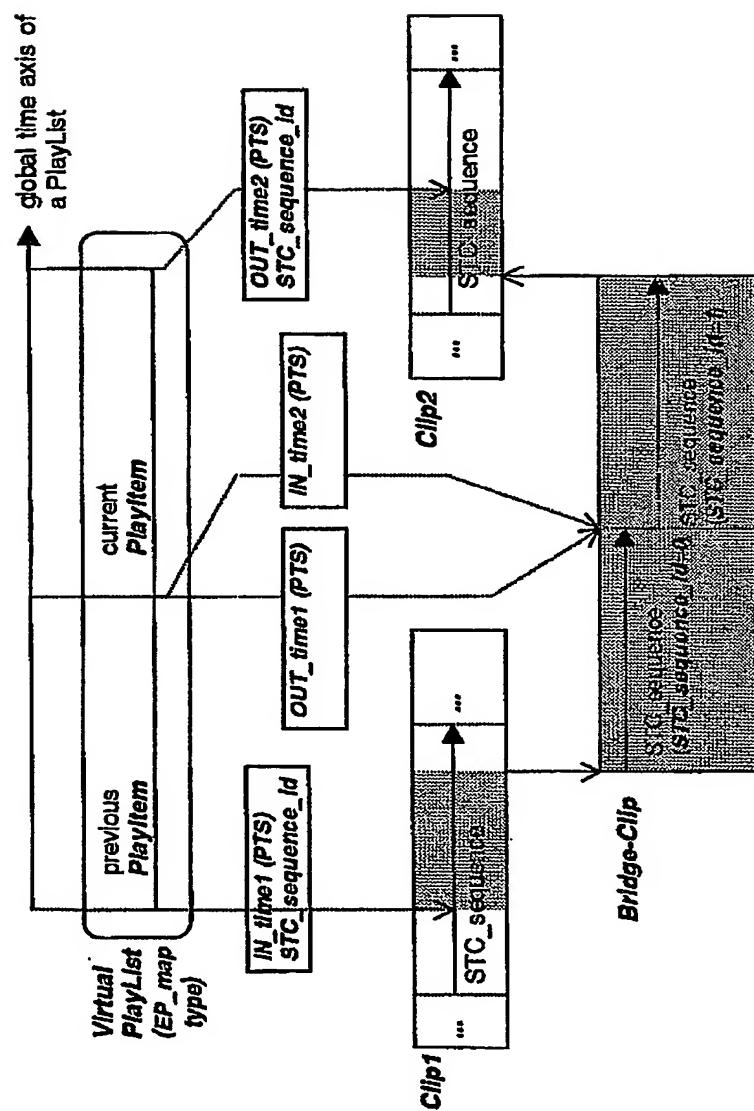
archive

【図29】



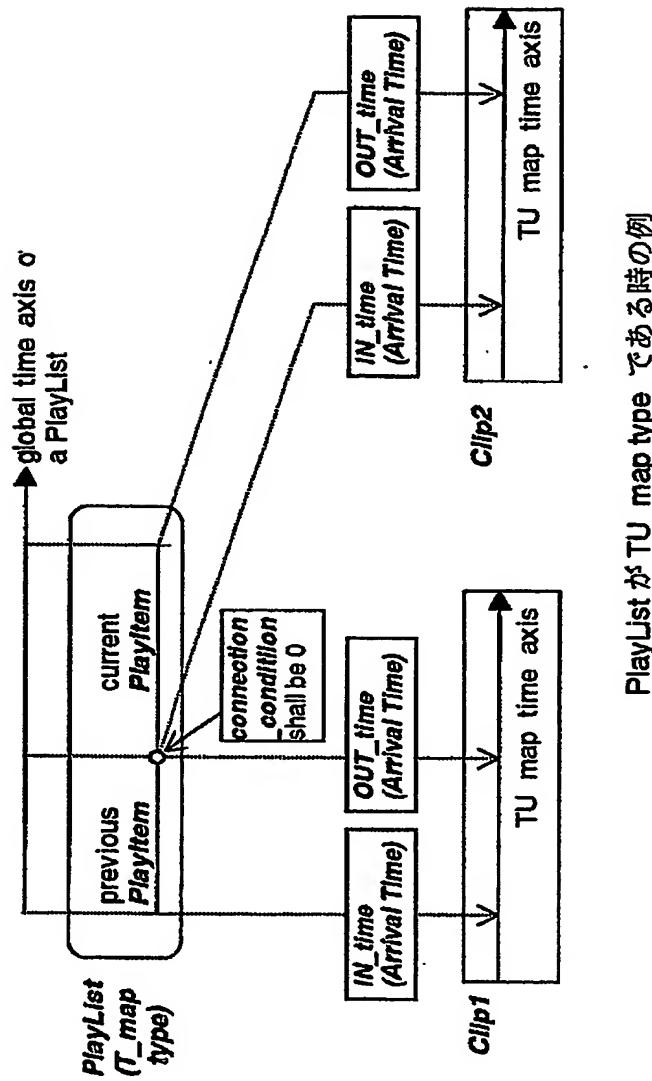
PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図30】



PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持つ時の例

【図31】

Playlist が TU_map type である時の例

【図32】

Syntax	No. of bits	Mnemonics
PlayItem()		
Clip	8*10	bslbf
Information	24	bslbf
file	8	uimsbf
name	32	uimsbf
reserved	32	uimsbf
STC	14	bslbf
sequence	2	bslbf
id		
IN		
time		
OUT		
time		
reserved		
connection		
condition		
if (<Virtual Playlist> {		
if (connection condition=='10') {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシナリオ

【図33】

CPI_type in the_PlayList()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対する 33 ビット長の PTS の上位 32 ビットを表示しなければならない。
TU_map type	IN_time は、TU_map_time_axis 上の時刻でなければならない。かつ、IN_time は、time_unit の精度に丸めて表さねばならない。IN_time は、次に示す等式により計算される。 $IN_time = TU_start_time \% 2^{32}$

IN_time

【図34】

CPI_type In the PlayList()	Semantics of OUT_time
EP_map type	<p>OUT_time は、次に示す等式によつて計算される。</p> <p>$\text{Presentation_end_TS}$ の値の上位 32 ビットを示さなければならない。</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>ここで、</p> <p>PTS_out は、PlayItem の中で最後のプレゼンテーションユニットに対する 33 ビット長の PTS である。</p> <p>AU_duration は、最後のプレゼンテーションユニットの 90kHz 単位の表示期間である。</p>
TU_map type	<p>OUT_time は、TU_map_time_axis 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さねばならない。</p> <p>OUT_time は、次に示す等式により計算される。</p> $\text{OUT_time} = \text{TU_start_time \% } 2^{32}$

OUT_time

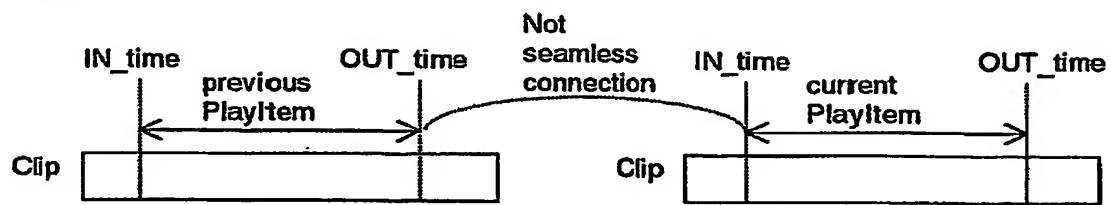
【図35】

<u>connection condition</u>	<u>meaning</u>
00	<ul style="list-style-type: none"> 先行する PlayItem と現在の PlayItem の接続は、シームレス再生の保証がなされていない。 PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされねばならない。
01	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、システムタイムベース (STC ベース) の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 この状態は、Virtual PlayList に対してだけ許される。 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。
11	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。

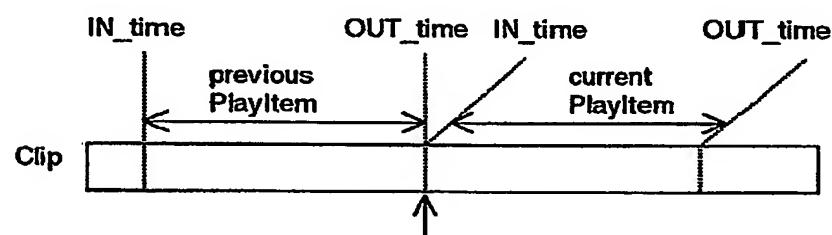
connection_condition

【図36】

(A)

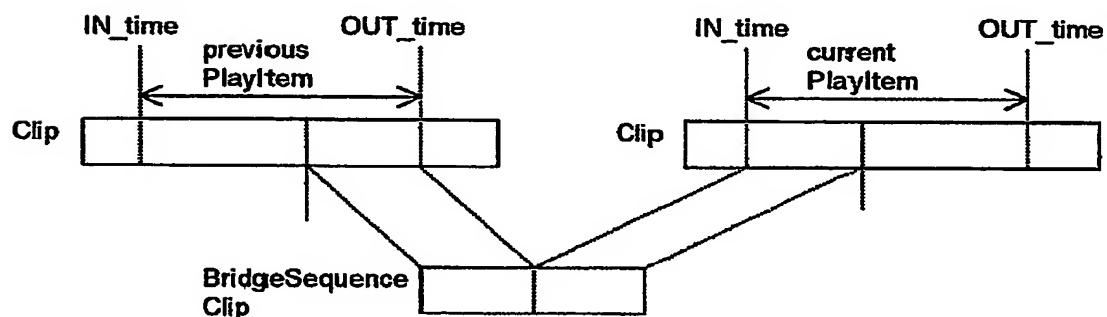


(B)



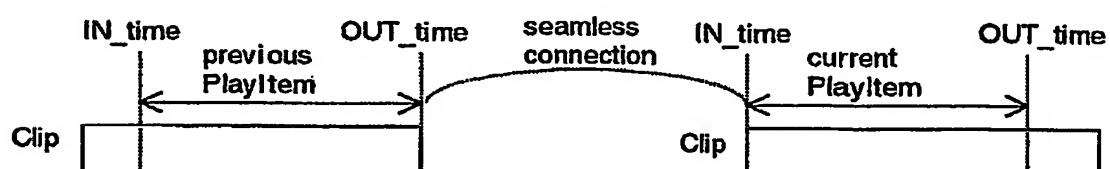
(C)

STC discontinuity
(system time base discontinuity)
connection_condition='01'



(D)

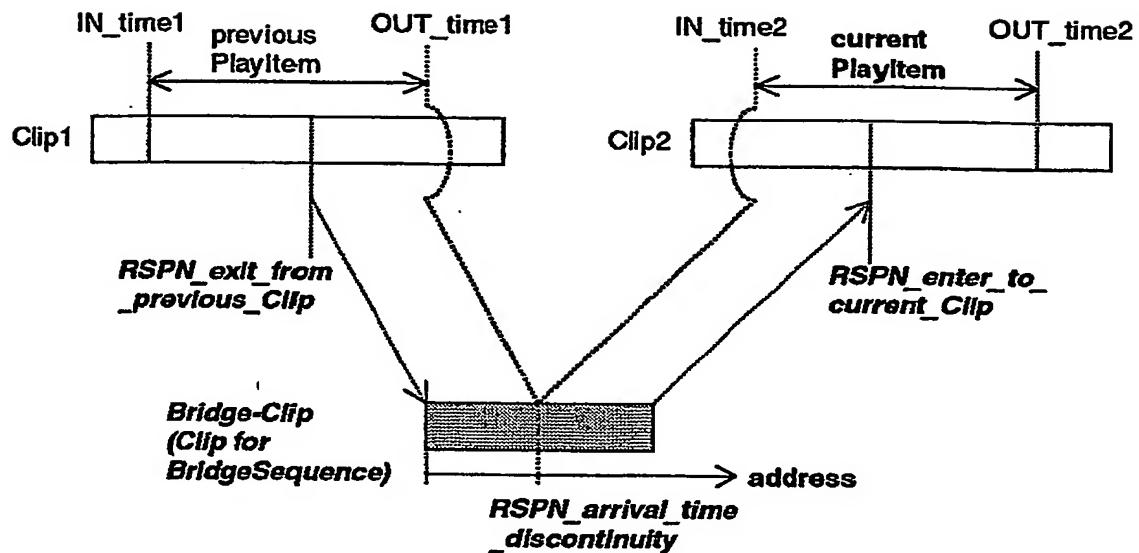
connection_condition='10'



connection_condition='11'

connection_condition の説明

【図37】

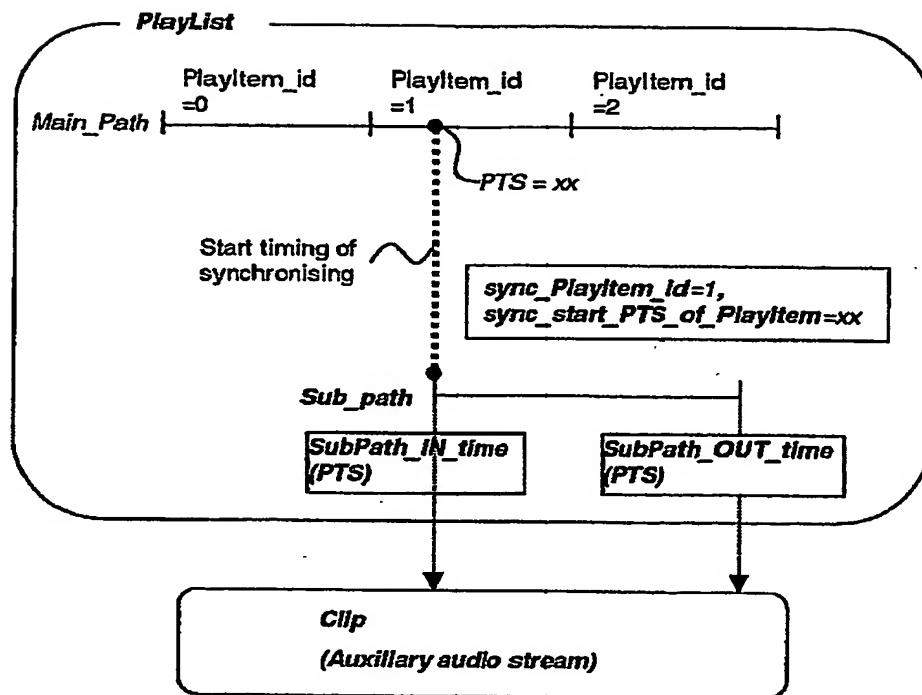


【図38】

Syntax	No. of bits	Mnemonics
BridgeSequenceInfo()		
Bridge Clip Information file name	8*10	bslbf
RSPN exit from previous Clip	32	ulmsbf
RSPN enter to current Clip	32	ulmsbf
}		

BridgeSequenceInfo のシンタクス

【图39】



【図40】

Syntax	No. of bits	Mnemonics
SubPlayItem{		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

SubPlayItem のシンタクス

【図4.1】

SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

【図4.2】

Syntax	No. of bits	No. of Mnemonics
PlayListMark()		
version_number	8*4	bslbf
length	32	ulmsbf
number_of_PlayList_marks	16	ulmsbf
for(i=0; i < number_of_PlayList_marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	ulmsbf
PlayItem_id	8	ulmsbf
reserved	24	ulmsbf
character_set	8	bslbf
name_length	8	ulmsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	ulmsbf
}		
}		

PlayListMarkのシントックス

【図4-3】

Mark_type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセツトすることができます、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03 - 0x8F	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図44】

CPI_type in the Playlist()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならぬ。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならぬ。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す等式により計算される。

$$\text{mark_time_stamp} = \text{TU_start_time \% } 2^{32}$$

mark_time_stamp

【図45】

Syntax	No. of bits	Mnemonics
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for(i=0;i<N1;i++) {		
padding_word	16	bslbf
}		
STC_Info()		
for(i=0;i<N2;i++) {		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0;i<N3;i++) {		
padding_word	16	bslbf
}		
CPI()		
for(i=0;i<N4;i++) {		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0;i<N5;i++) {		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

zzzzz.clpi のシンタクス

【図46】

Syntax	No. of bits	Mnemonics
ClipInfo()		
version number	8*4	bslbf
length	32	uimsbf
Clip stream type	8	bslbf
offset SPN	32	uimsbf
TS recording rate	24	uimsbf
reserved	8	bslbf
record time and date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time controlled flag	1	bslbf
TS average rate	24	uimsbf
if (Clip stream type==1) // Bridge-Clip AV stream		
RSPN arrival time discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved for system use	144	bslbf
reserved	11	bslbf
is format identifier valid	1	bslbf
is original network ID valid	1	bslbf
is transport stream ID valid	1	bslbf
is servece ID valid	1	bslbf
is country code valid	1	bslbf
format identifier	32	bslbf
original network ID	16	uimsbf
transport stream ID	16	uimsbf
servece ID	16	uimsbf
country code	24	bslbf
stream format name	16*8	bslbf
reserved for future use	256	bslbf
}		

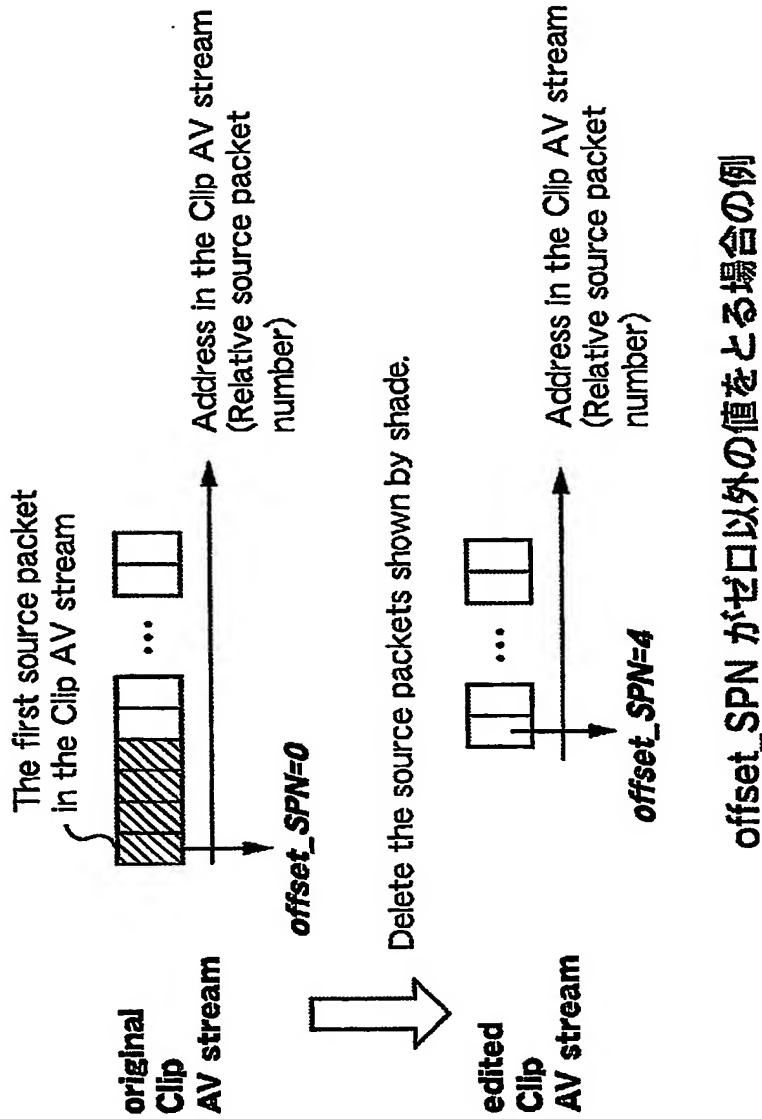
ClipInfo のシンタクス

【図4.7】

Clip_stream_type	meaning
0	Clip AVストリーム
1	Bridge-Clip AVストリーム
2 - 255	Reserved

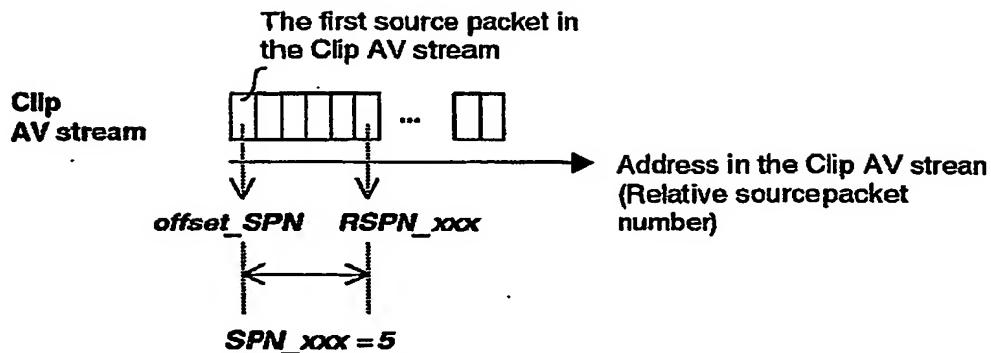
Clip_stream_type

【図4.8】



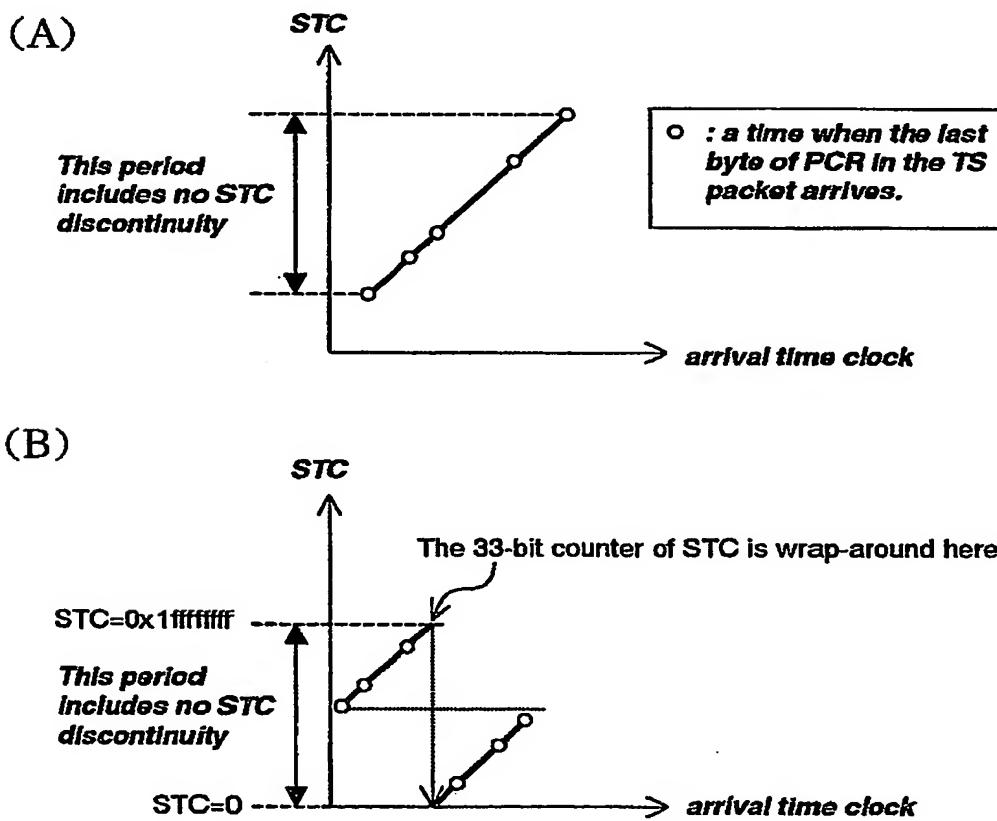
offset_SPN がゼロ以外の値をとる場合の例

【図49】

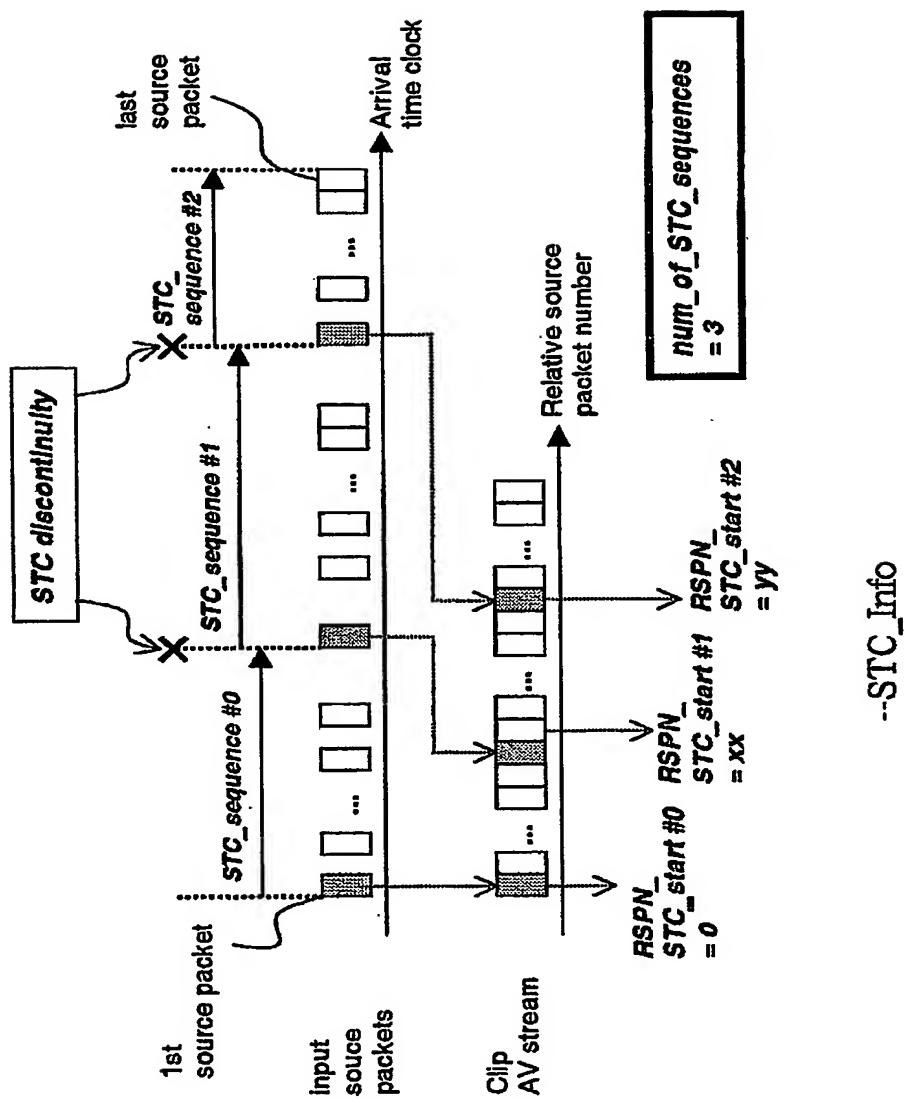


AVストリームでの offset_SPN と相対ソースパケット番号 (RSPN_xxx) の間の関係

【図50】



【図51】

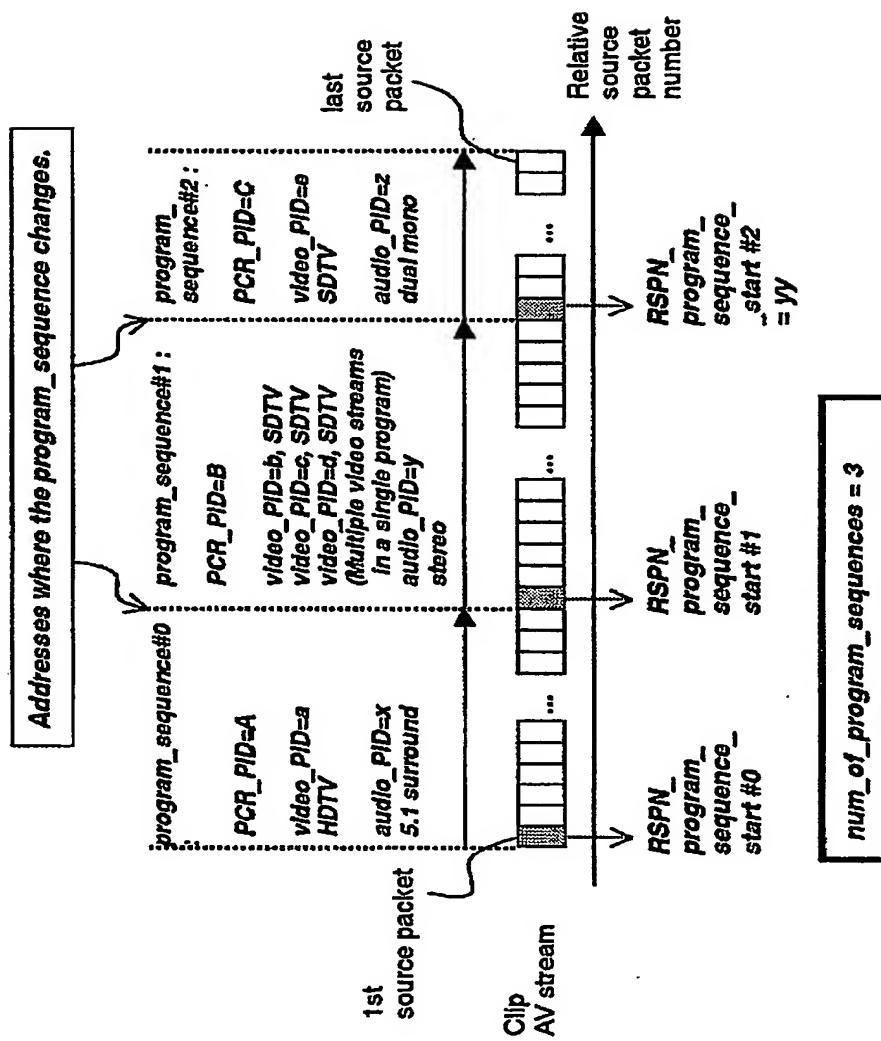


【図52】

Syntax	No. of bits	Mnemonics
<code>STC_Info() {</code>		
<code> version_number</code>	<code>8*4</code>	<code>bslbf</code>
<code> length</code>	<code>32</code>	<code>ulmsbf</code>
<code> if (length != 0) {</code>		
<code> reserved</code>	<code>8</code>	<code>bslbf</code>
<code> num_of_STC_sequences</code>	<code>8</code>	<code>ulmsbf</code>
<code> for(STC_sequence_id=0;</code>		
<code> STC_sequence_id < num_of_STC_sequences;</code>		
<code> STC_sequence_id++) {</code>		
<code> reserved</code>	<code>32</code>	<code>bslbf</code>
<code> RSPN_STC_start</code>	<code>32</code>	<code>ulmsbf</code>
<code> }</code>		
<code> }</code>		
<code>}</code>		

STC_Info のシルタクス

【図53】



ProgramInfo の例

【図54】

Syntax	No. of bits	Mnemonics
ProgramInfo()		
version number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for(i=0; i<number_of_program_sequences; i++) {		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0; k<number_of_videos; k++) {		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0; k<number_of_audios; k++) {		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		

ProgramInfo のシンタクス

【図55】

Syntax	No. of bits	Mnemonics
<code>VideoCodingInfo() {</code>		
<code> video format</code>	8	uimbsbf
<code> frame rate</code>	8	uimbsbf
<code> display_aspect_ratio</code>	8	uimbsbf
<code> reserved</code>	8	bslbf
<code>}</code>		

VideoCodingInfo のシンタクス

【図56】

video_format	Meaning
0	480i
1	576i
2	480p (including 640x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

video_format

【図57】

frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94 ..)
8	60
9 - 254	reserved
255	No information

frame_rate

【図58】

<u>display_aspect_ratio</u>	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

display_aspect_ratio

【図59】

Syntax	No. of bits	Mnemonics
AudioCodingInfo()		
audio_coding	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

AudioCodingInfo のシンタクス

【図60】

<u>audio_coding</u>	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

audio_coding

【図61】

<u>audio_component_type</u>	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

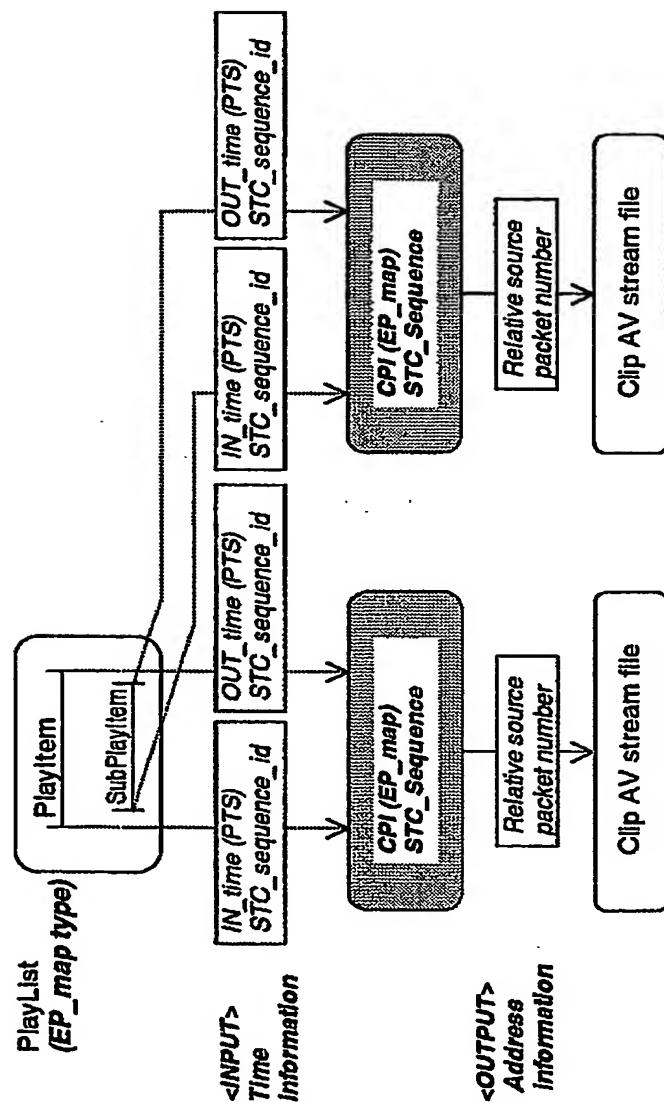
audio_component_type

【図62】

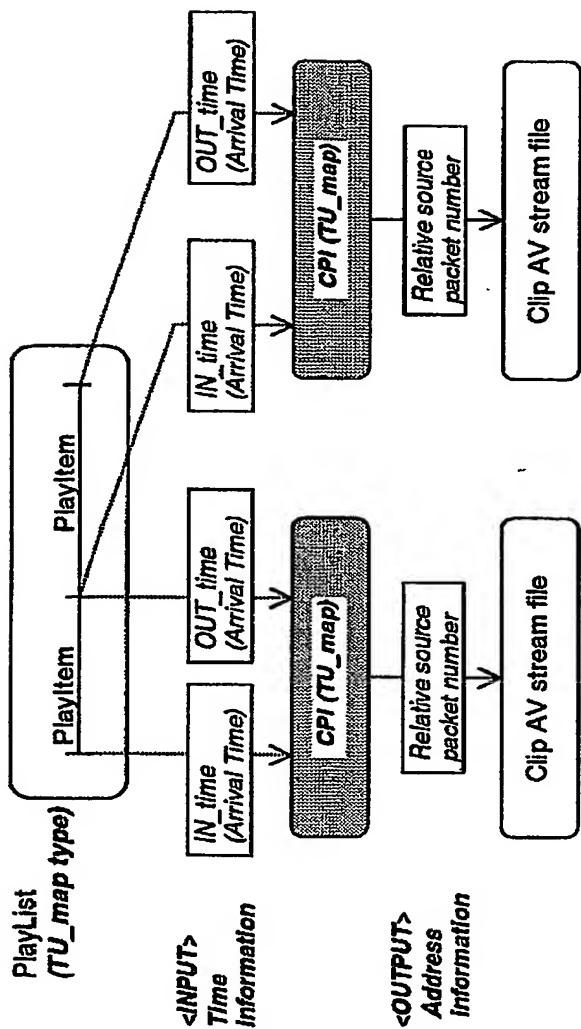
<u>sampling_frequency</u>	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【図63】



【図64】



【図65】

Syntax	No. of bits	Mnemonics
CPI{		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type == 0)		
EP_map()		
else		
TU_map()		
}		

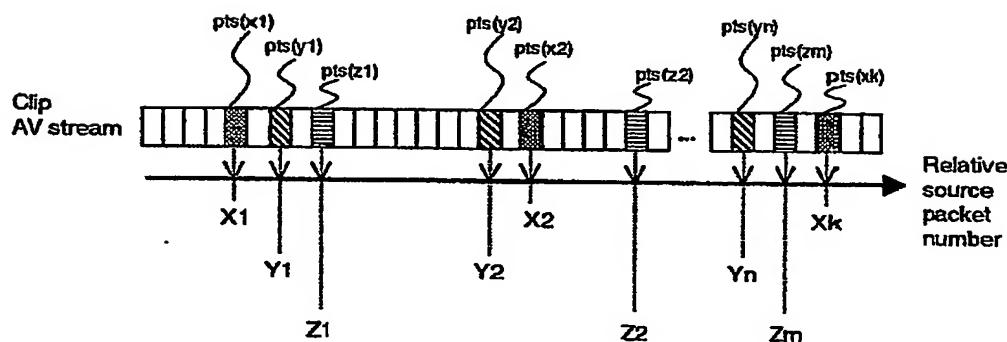
CPI のシンタクス

【図66】

CPI_type	Meaning
0	EP map type
1	TU map type

CPI_type の意味

【図67】



■ : source packets that includes the first byte of the *sequence header*.
video_PID=x

■ : source packets that includes the first byte of the *sequence header*.
video_PID=y

■ : source packets that includes the first byte of the *sequence header*.
video_PID=z

EP_map

number_of_stream_PIDs=3

EP_map_for_one_stream_PID(0)

stream_PID(0) = x
num_EP_entries(0) = k

PTS_EP_start	RSPN_EP_start
pts(x1)	X1
pts(x2)	X2
...	...
pts(xk)	Xk

EP_map_for_one_stream_PID(1)

stream_PID(1) = y
num_EP_entries(1) = n

PTS_EP_start	RSPN_EP_start
pts(y1)	Y1
pts(y2)	Y2
...	...
pts(yn)	Yn

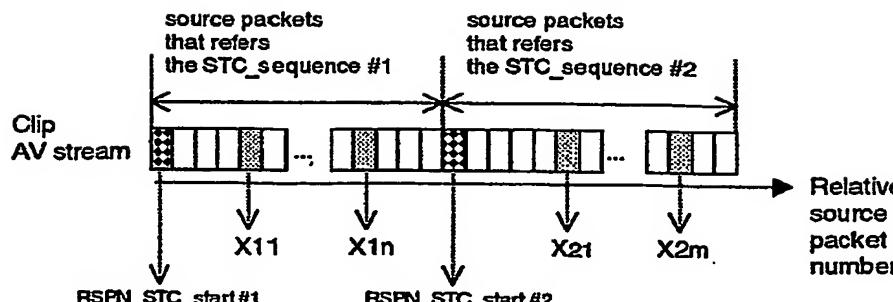
EP_map_for_one_stream_PID(2)

stream_PID(2) = z
num_EP_entries(2) = m

PTS_EP_start	RSPN_EP_start
pts(z1)	Z1
pts(z2)	Z2
...	...
pts(zm)	Zm

ビデオの EP_map の例

【図68】

*EP_map_for_one_stream_PID**video_PID=x*

<i>PTS_EP_start</i>	<i>RSPN_EP_start</i>
<i>pts(x11)</i> ...	<i>X11</i> ...
<i>pts(x1n)</i>	<i>X1n</i>

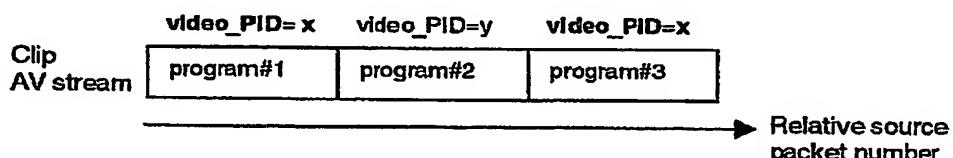
These data belong to the STC_sequence #1

boundary

RSPN_STC_start #2 < X21

These data belong to the STC_sequence #2

【図69】

*EP_map**number_of_stream_PIDs=3**EP_map_for_one_stream_PID(0)**stream_PID(0) = x*
...*EP_map_for_one_stream_PID(1)**stream_PID(1) = y*
...*EP_map_for_one_stream_PID(2)**stream_PID(2) = x*
...

【図70】

Syntax	No. of bits	Mnemonics
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k< number_of_stream_PIDs ;k++){		
stream_PID (k)	16	bslbf
num_EP_entries (k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for(i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k< number_of_stream_PIDs ;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for(i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

【図71】

EP_type	Meaning
0	video
1	audio
2 - 15	reserved

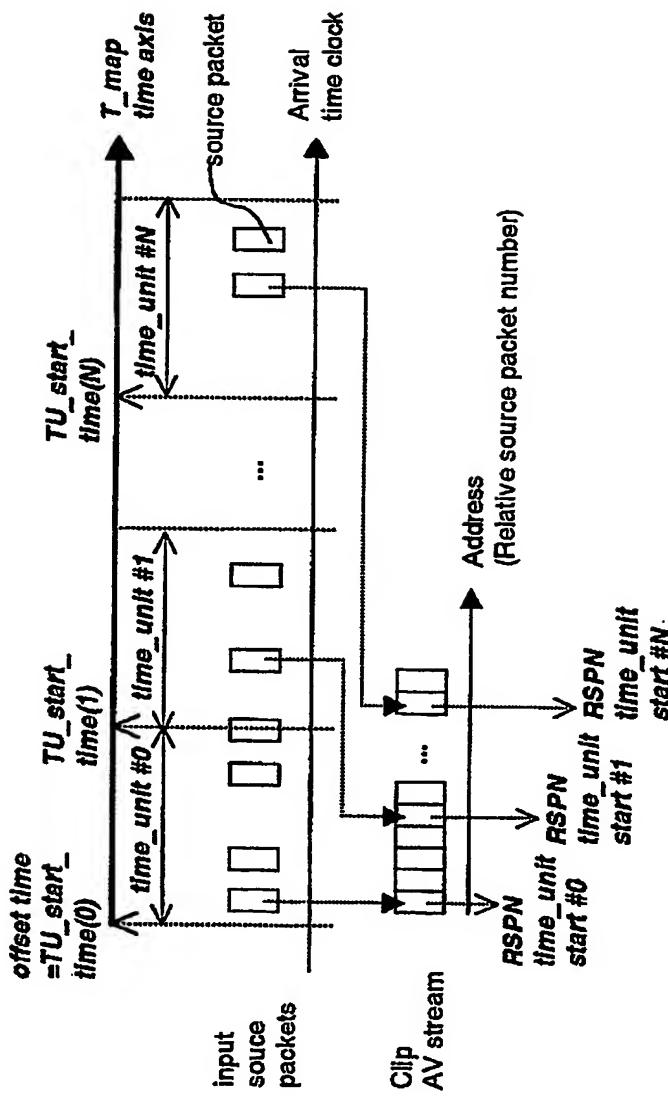
EP_type Values

【図72】

Syntax	No. of bits	Mnemonics
EP_map_for_one_stream_PID(N){		
for (i=0; i < N; i++) {		
PTS_EP_start	32	uimbsbf
RSPN_EP_start	32	uimbsbf
}		
}		

EP_map_for_one_stream_PID のシンタクス

【図73】



【図74】

Syntax	No. of bits	Mnemonics
TU_map{		
offset_time	32	bslbf
time_unit_size	32	ulmsbf
number_of_time_unit_entries	32	ulmsbf
for (k=0; k<number_of_time_unit_entries; k++)		
RSPN_time_unit_start	32	ulmsbf
}		

TU_map のシンタクス

【図75】

Syntax	No. of bits	No. of Mnemonics
ClipMark()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for(i=0; i < number_of_Clip_marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

ClipMarkのシンタクス

【図76】

Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlaylistMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーン切替ポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

【図77】

CPI_type in the CPI0	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならぬ。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならぬ。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す等式により計算される。

$$\text{mark_time_stamp} = \text{TU_start_time \% } 2^{32}$$

mark_type_stamp

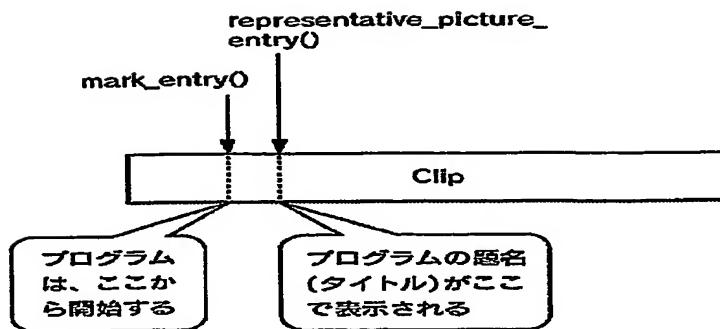
【図78】

Syntax	No. of bits	Mnemonics
ClipMark()		
version_number	8 [~] 4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
reserved_for_MakerID	16	bslbf
mark_entry()		
representative_picture_entry()		
ref_thumbnail_index	16	uimsbf
}		
}		

【図79】

Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーン開始ポイントを示すマーク点
0x93	Scene-end mark	シーン終了ポイントを示すマーク点
0x94	CM-start mark	CM 開始ポイントを示すマーク点
0x95	CM-end mark	CM 終了ポイントを示すマーク点
0x96 - 0xBF	DVR フォーマットが、 ClipMark を将来、拡張する時のために予約されている	
0xC0 - 0xFF	メーカー独自のアプリケーションで利用するマークに割り当て可能	

【図80】



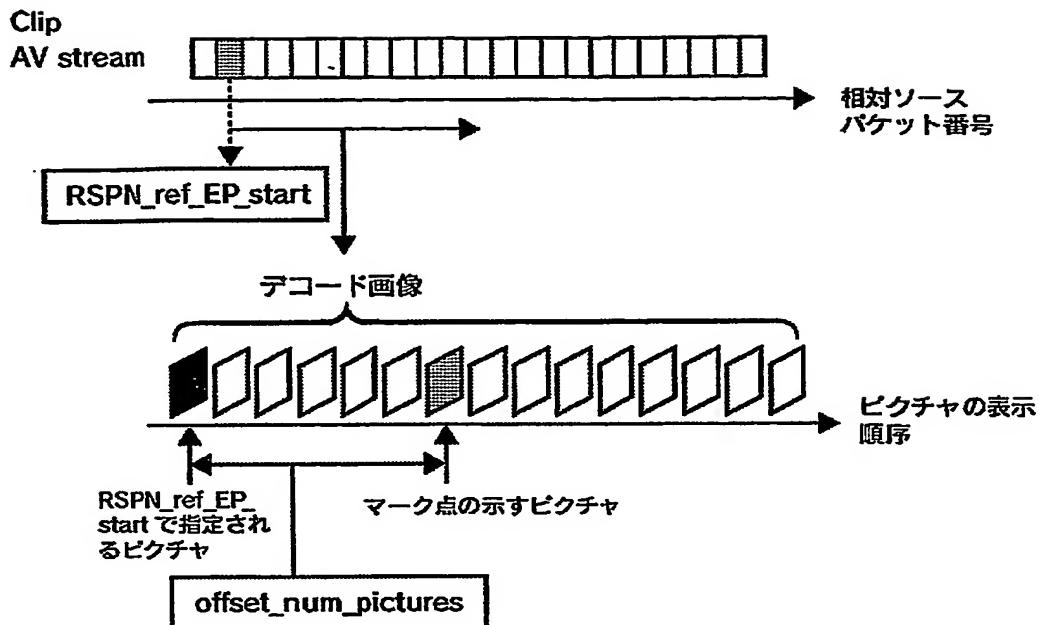
【図81】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
mark_time_stamp	32	uimsbf
STC_sequence_Id	8	uimsbf
reserved	24	bslbf
}		

【図82】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

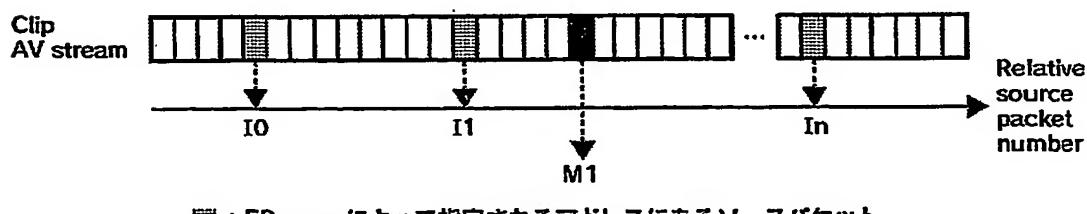
【図83】



【図84】

Syntax	No. bits	of	Mnemonics
<code>mark_entry() / representative_picture_entry()</code>			
RSPN_mark_point	32		<code>uimsbf</code>
}			

【図85】



■ : EP_map によって指定されるアドレスにあるソースパケット。
このソースパケットから、Iピクチャが開始している。

■ : ClipMark で指定されるアドレスにあるソースパケット。
このソースパケットからマークで指定されるピクチャが開始している。

【図86】

Syntax	No. of bits	Mnemonics
menu.thmb / mark.thmb {		
reserved	256	bs bf
Thumbnail()		
for(i=0; i<N1; i++)		
padding_word	16	bs bf
}		

menu.thmb & mark.thmb のシンタクス

【図87】

Syntax	Bits	Mnemonics
Thumbnail0 {		
version_number	8*4	char
length	32	ulmsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	ulmsbf
tn_block_size	16	ulmsbf
number_of tn_blocks	16	ulmsbf
reserved	16	bslbf
for(l = 0; l < number_of thumbnails; l++) {		
thumbnail_index	16	ulmsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	ulmsbf
start tn_block_number	16	ulmsbf
x_picture_length	16	ulmsbf
y_picture_length	16	ulmsbf
reserved	16	ulmsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k = 0; k < number_of tn_blocks; k++) {		
tn_block	tn_block_size*	1024*8
}		
}		

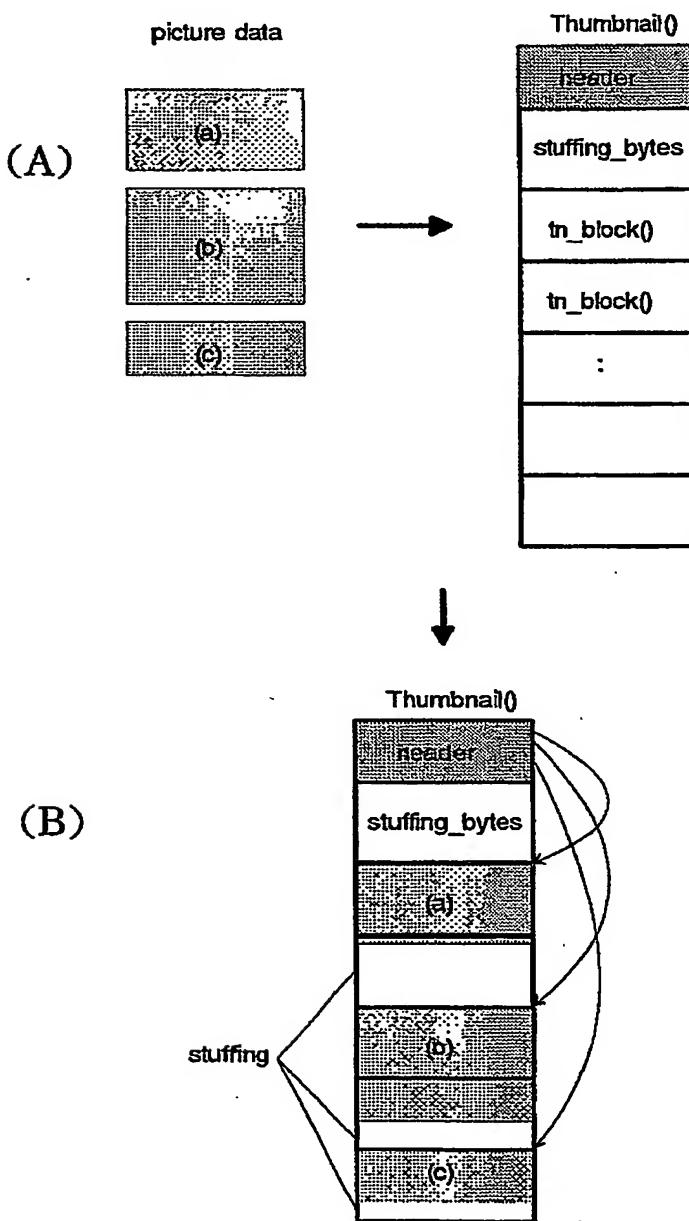
Thumbnail0シシタクス

【図88】

<u>Thumbnail picture format</u>	<u>Meaning</u>
<i>0x00</i>	<i>MPEG-2 Video I-picture</i>
<i>0x01</i>	<i>DCF (restricted JPEG)</i>
<i>0x02</i>	<i>PNG</i>
<i>0x03-0xff</i>	<i>reserved</i>

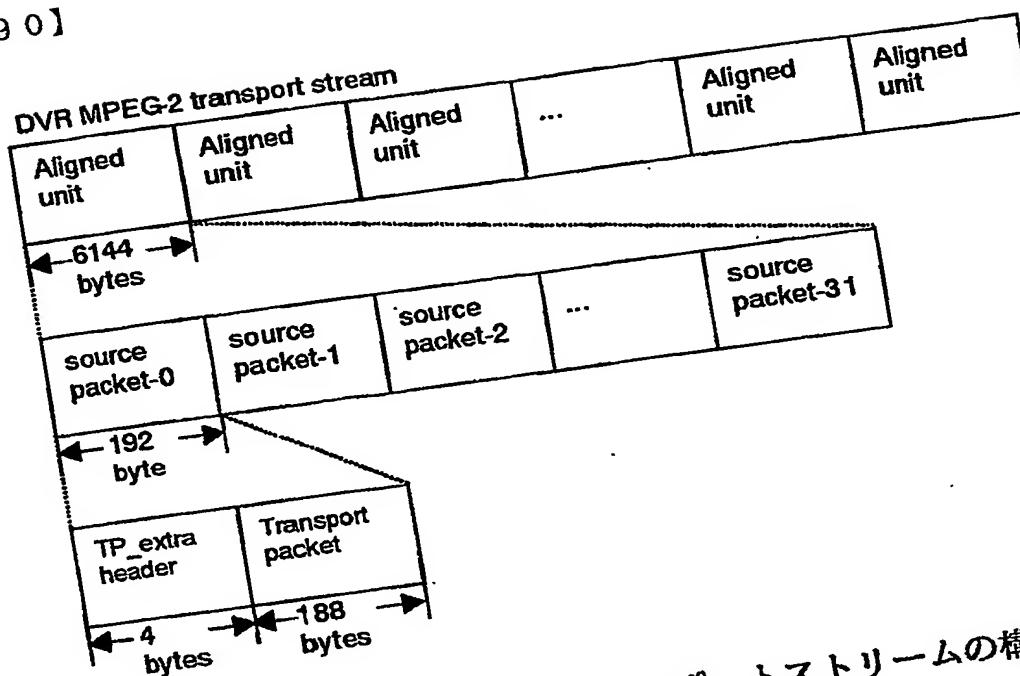
thumbnail picture format

【図 89】



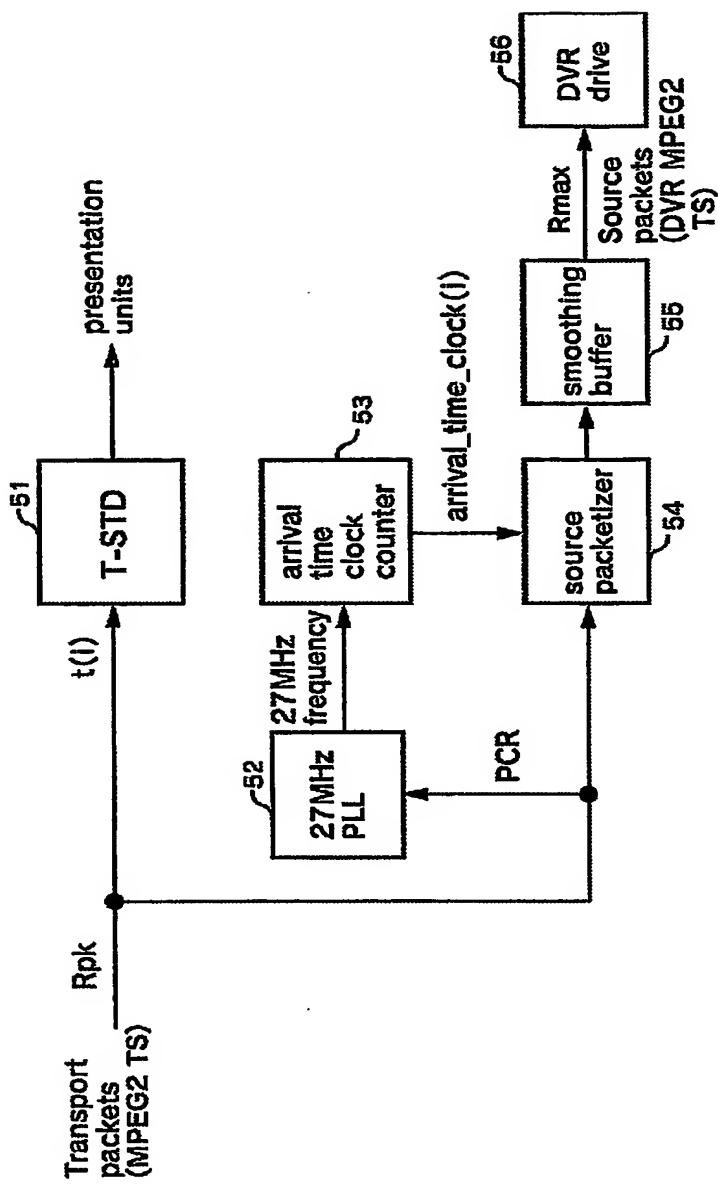
特2000-268043

【図90】



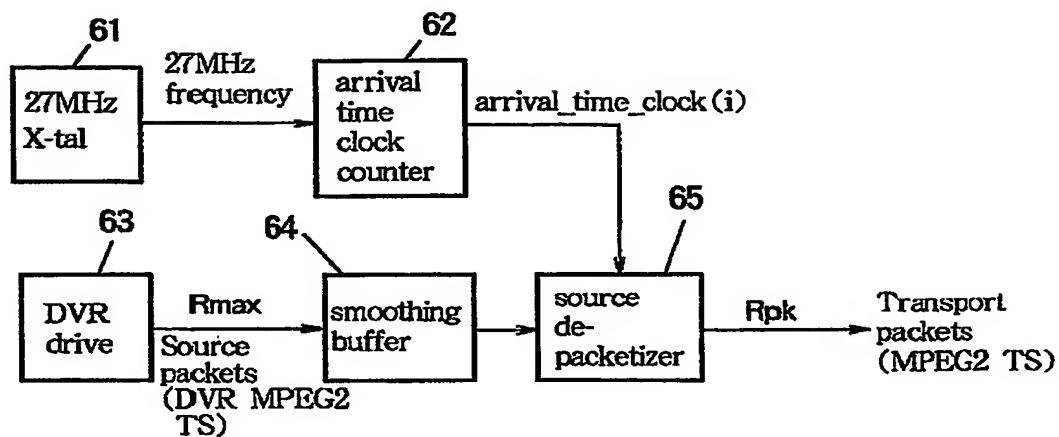
DVR MPEG-2 トランSPORTストリームの構造

【図91】



DVR MPEG-2 トランSPORTストリームのレコーダモデル

【図92】



DVR MPEG-2 トランスポートストリームのプレーヤモデル

[图93]

Syntax	No. of bits	Mnemonics
source_packet()		
TP extra_header()		
transport_packet()		

source packet

【図94】

Syntax	No. of bits	Mnemonics
TP extra_header{		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

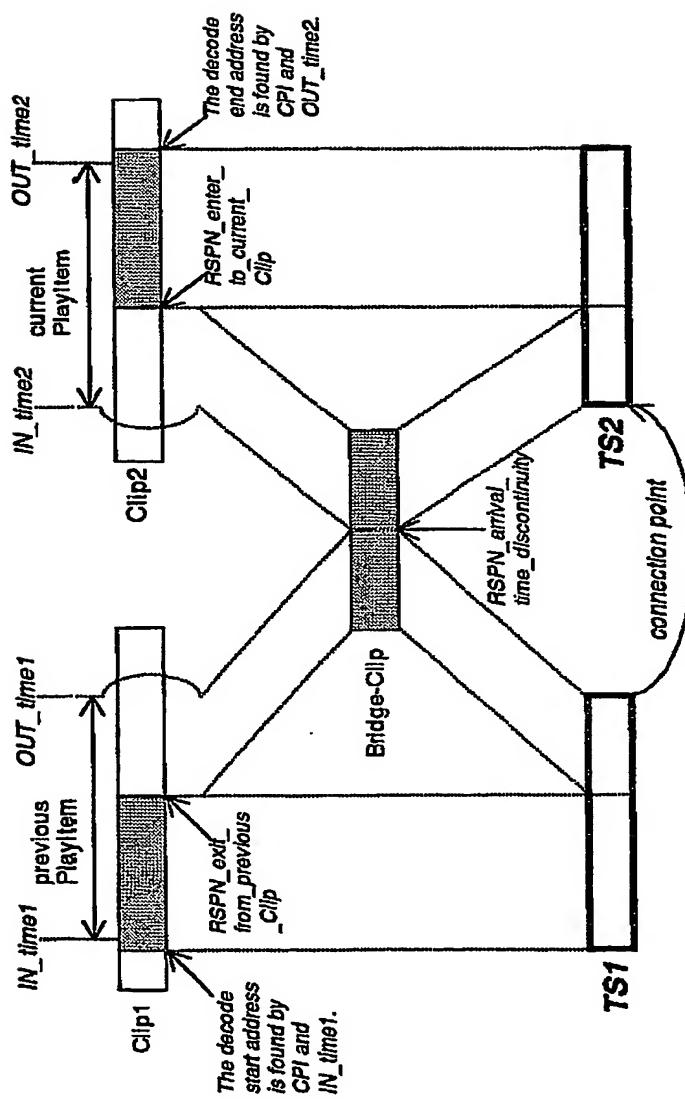
TP_extra_header

【図95】

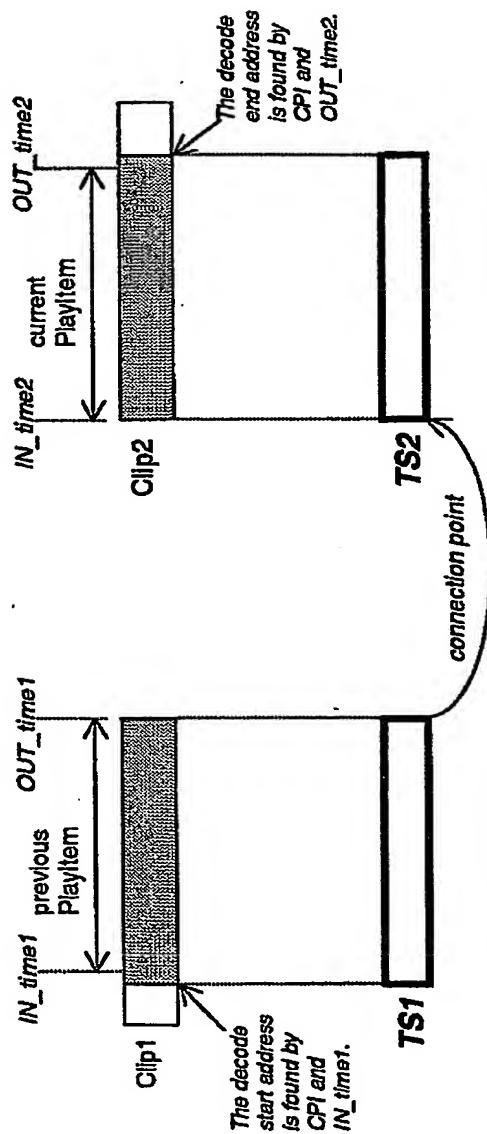
copy_permission indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

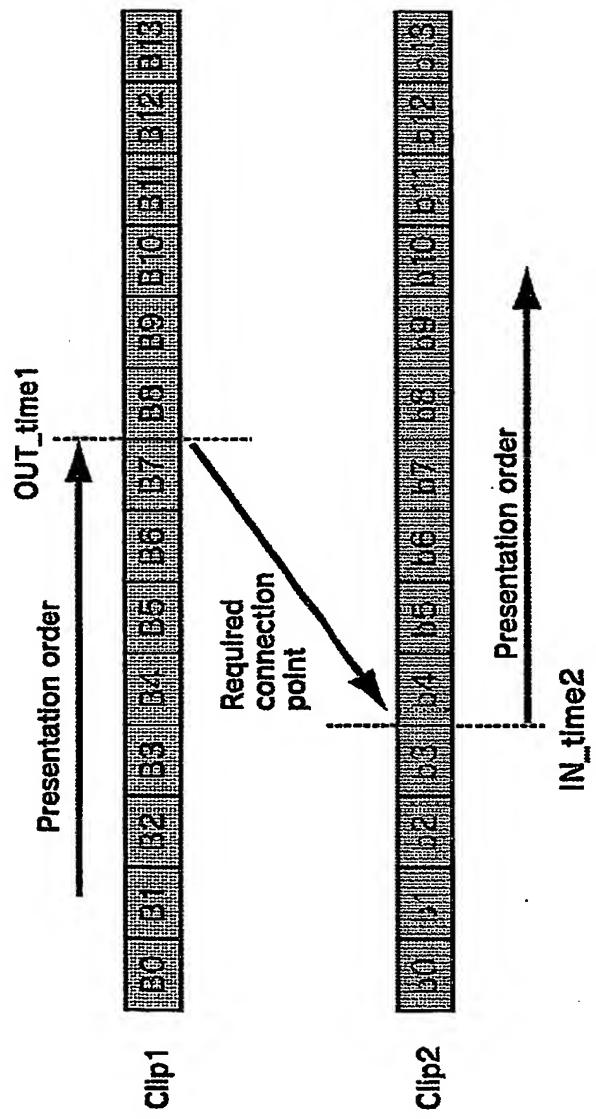
【図96】



【図97】

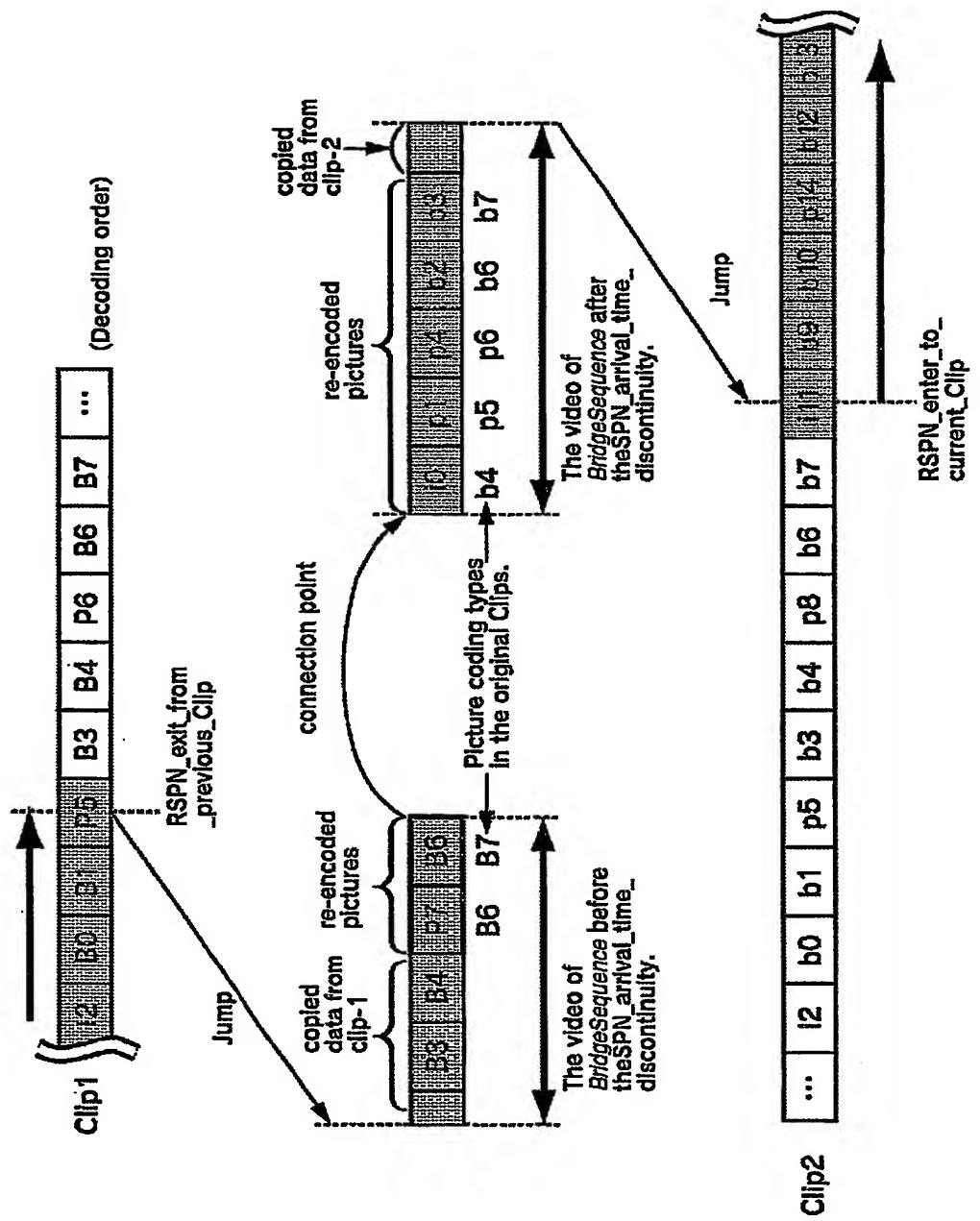


【図98】



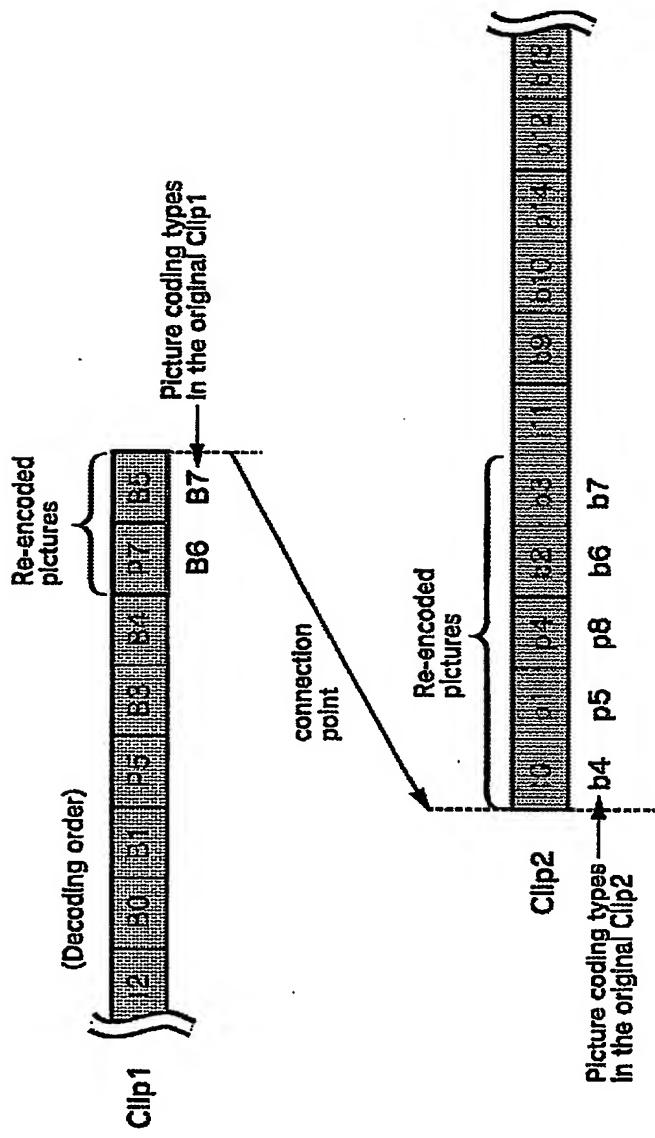
ピクチャの表示順序で示すシームレス接続の例

【図99】



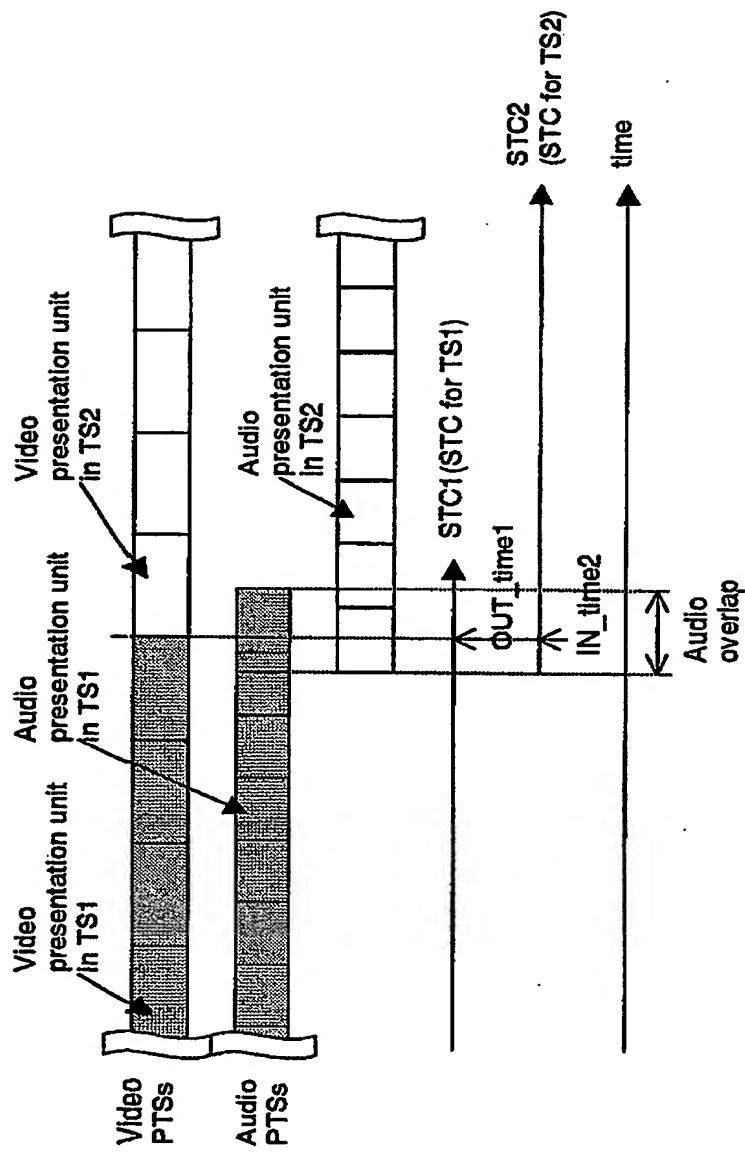
BridgeSequence を使用してシームレス接続を実現する例1

【図100】

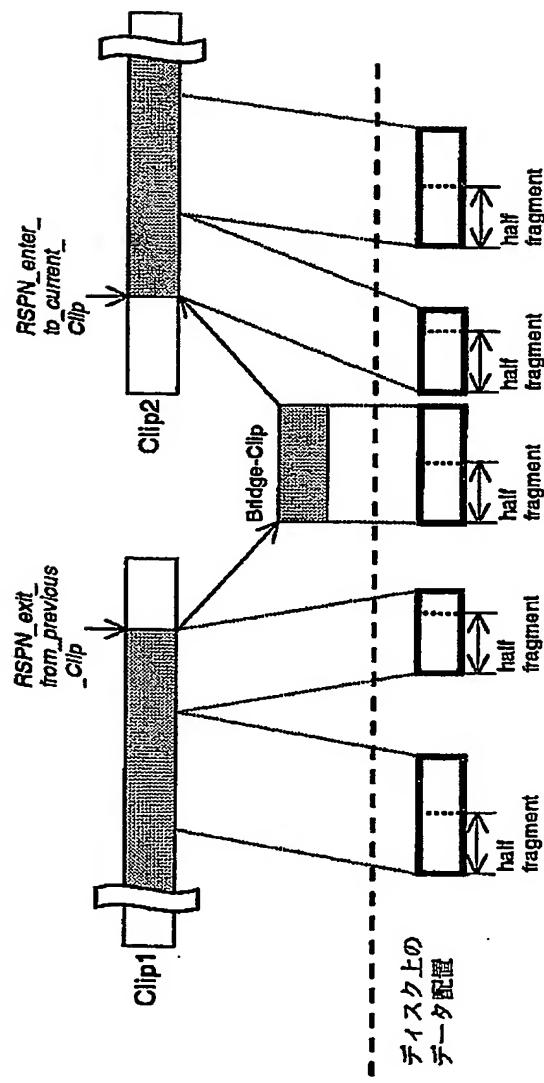


BridgeSequence を使用しないでシームレス接続を実現する例2

【図101】

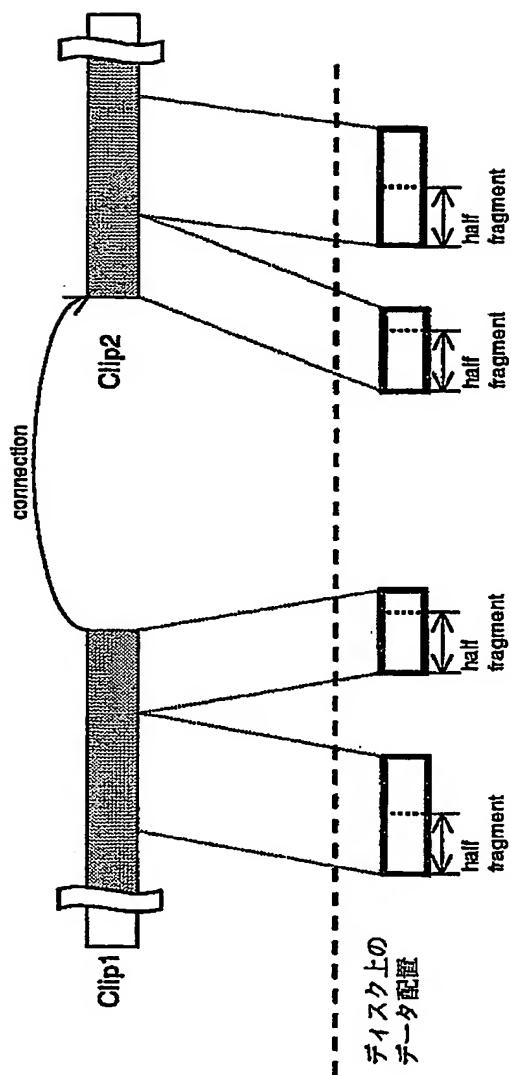


【図102】



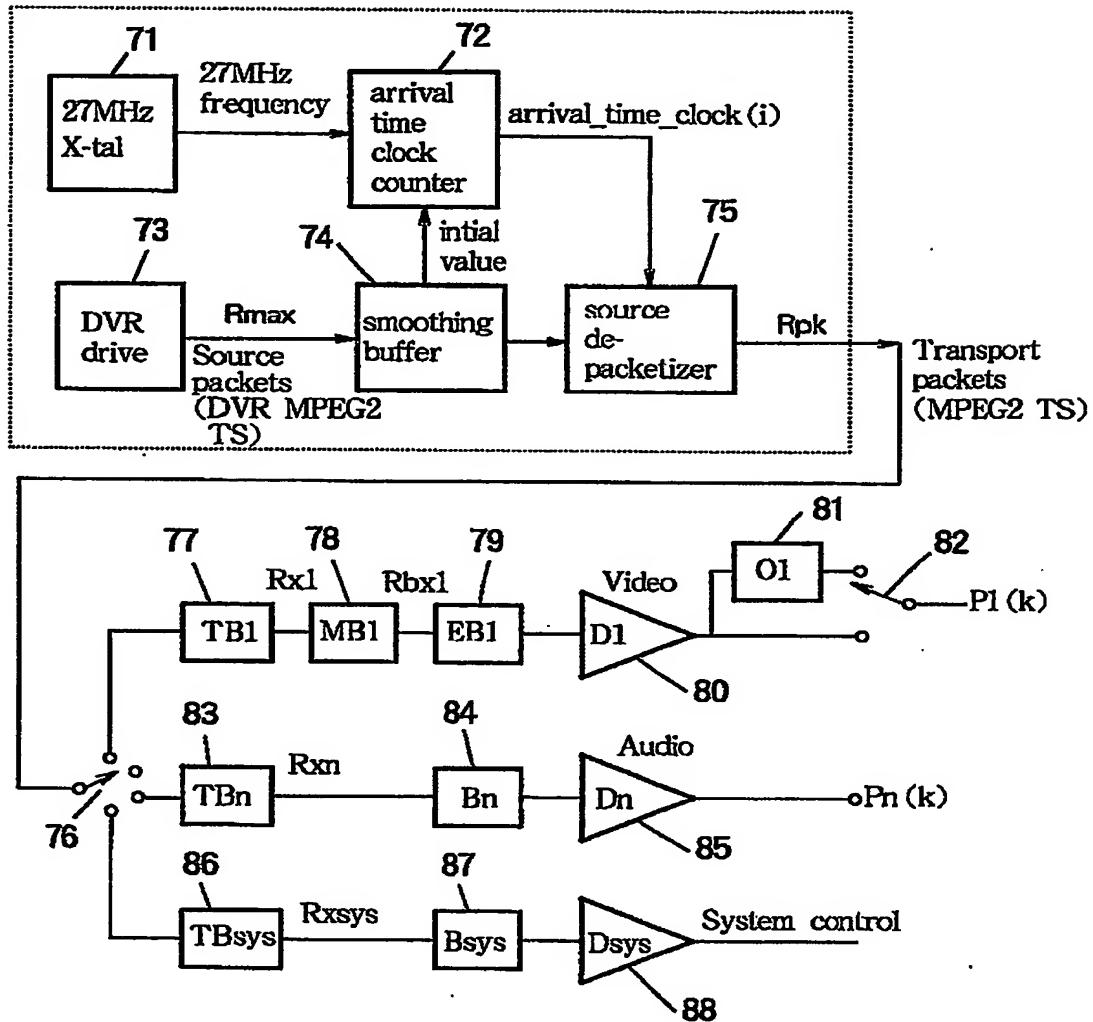
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図103】

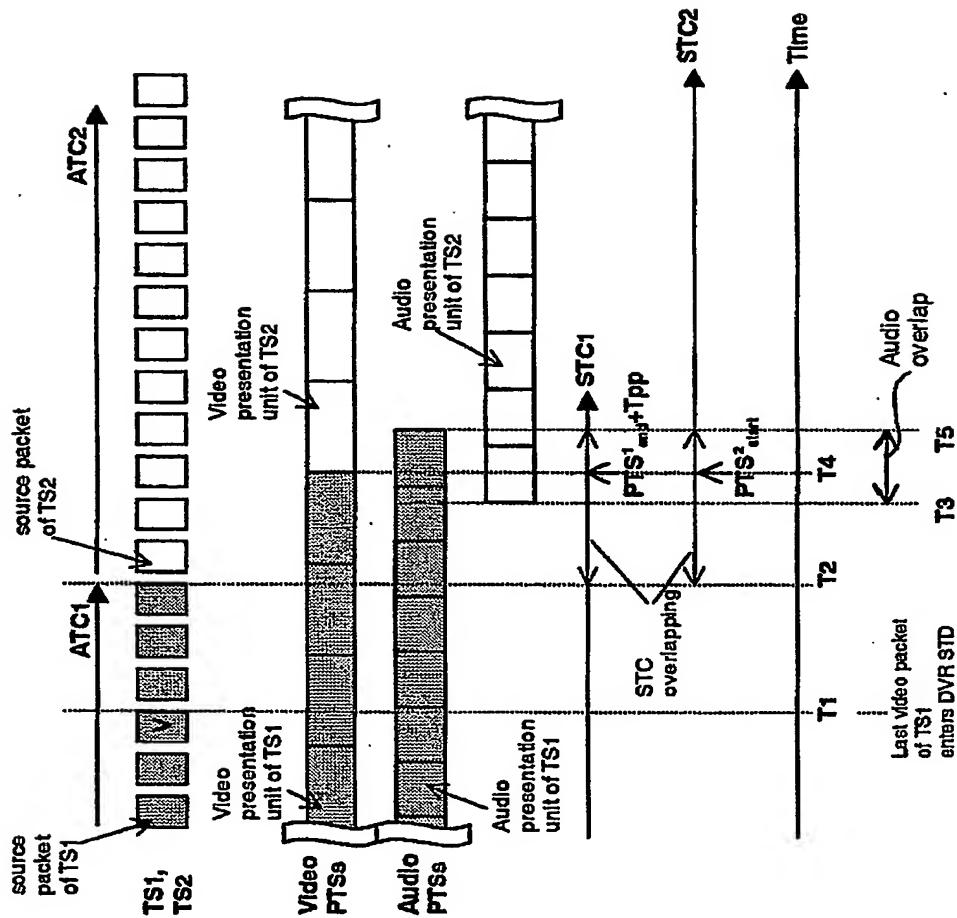


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

【図104】

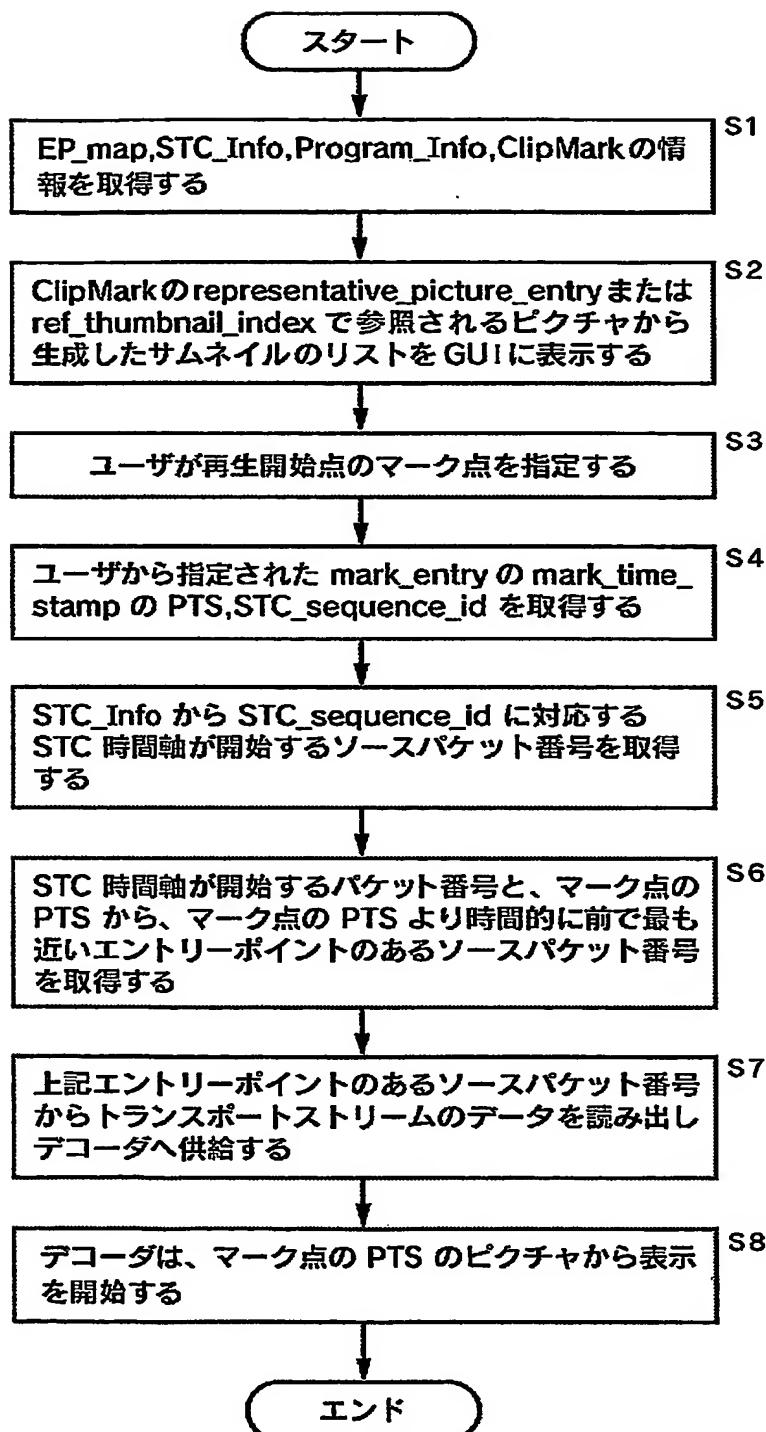


【図105】

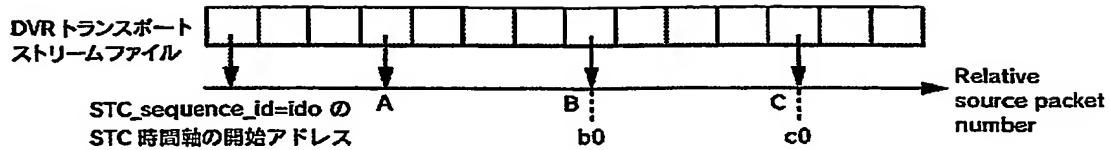


あるAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

【図106】



【図107】



【図108】

EP_map

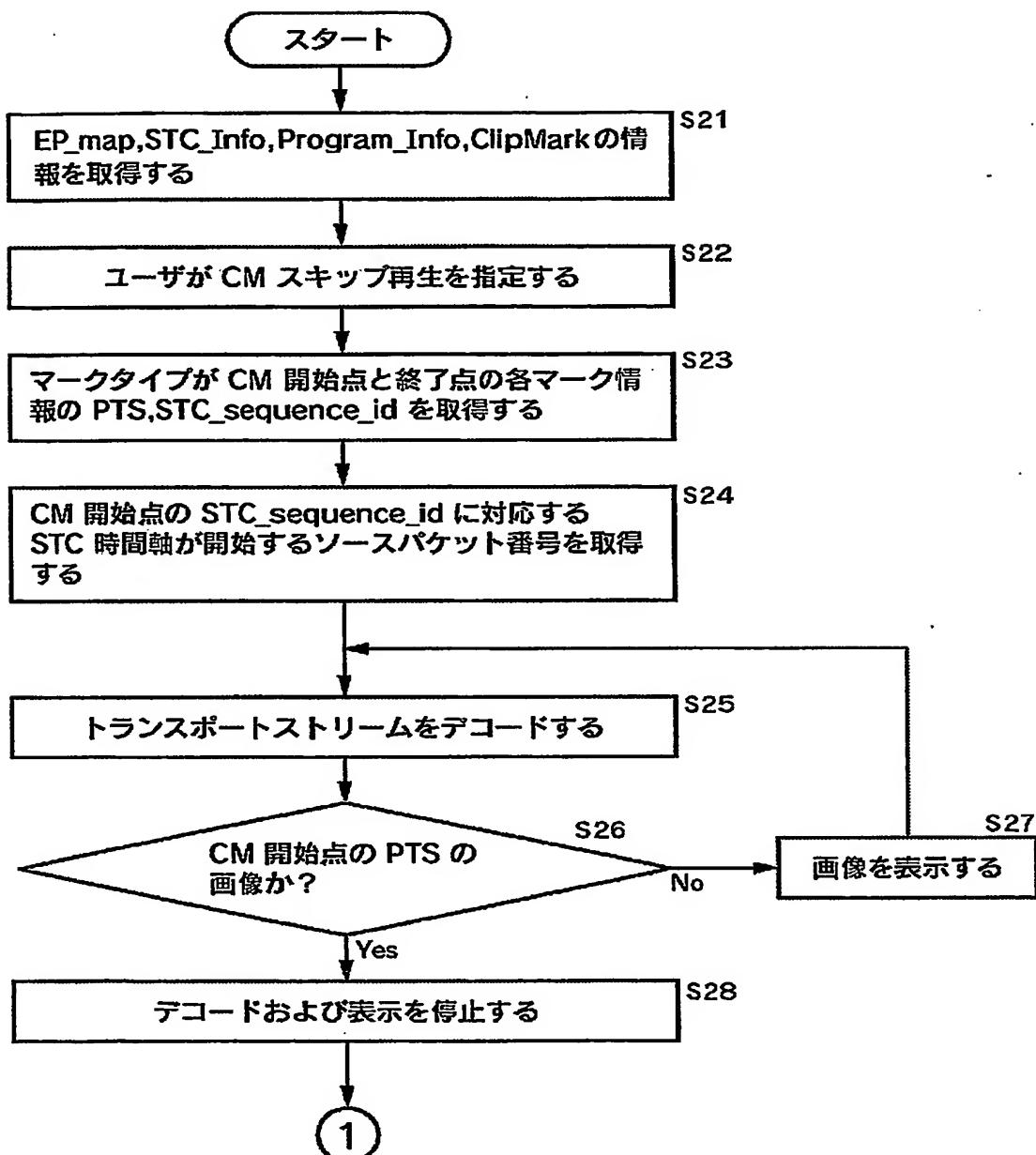
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

【図109】

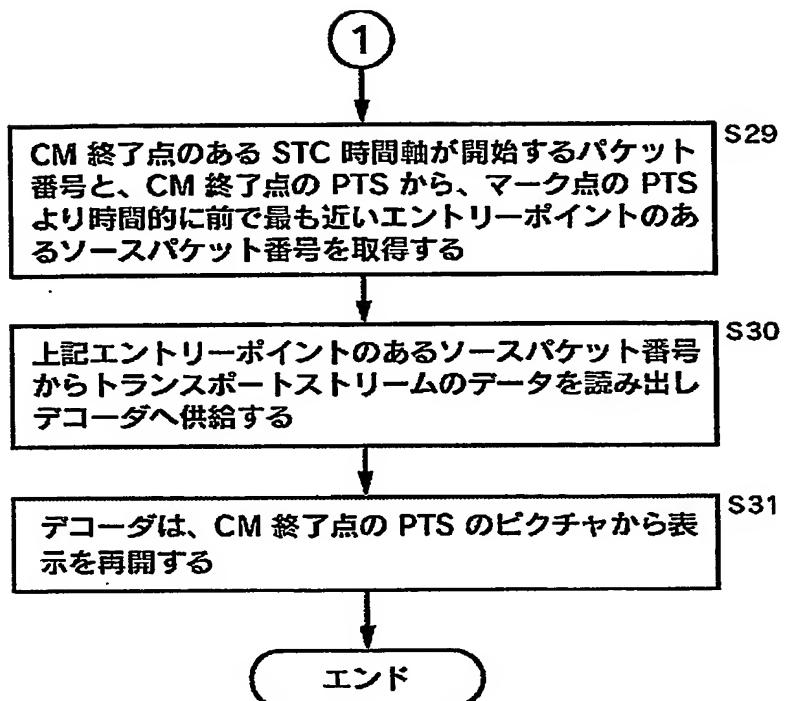
ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CMstart)	PTS(b0)	id0	PTS(b0)	id0
0x95(CMend)	PTS(c0)	id0	PTS(c0)	id0
...

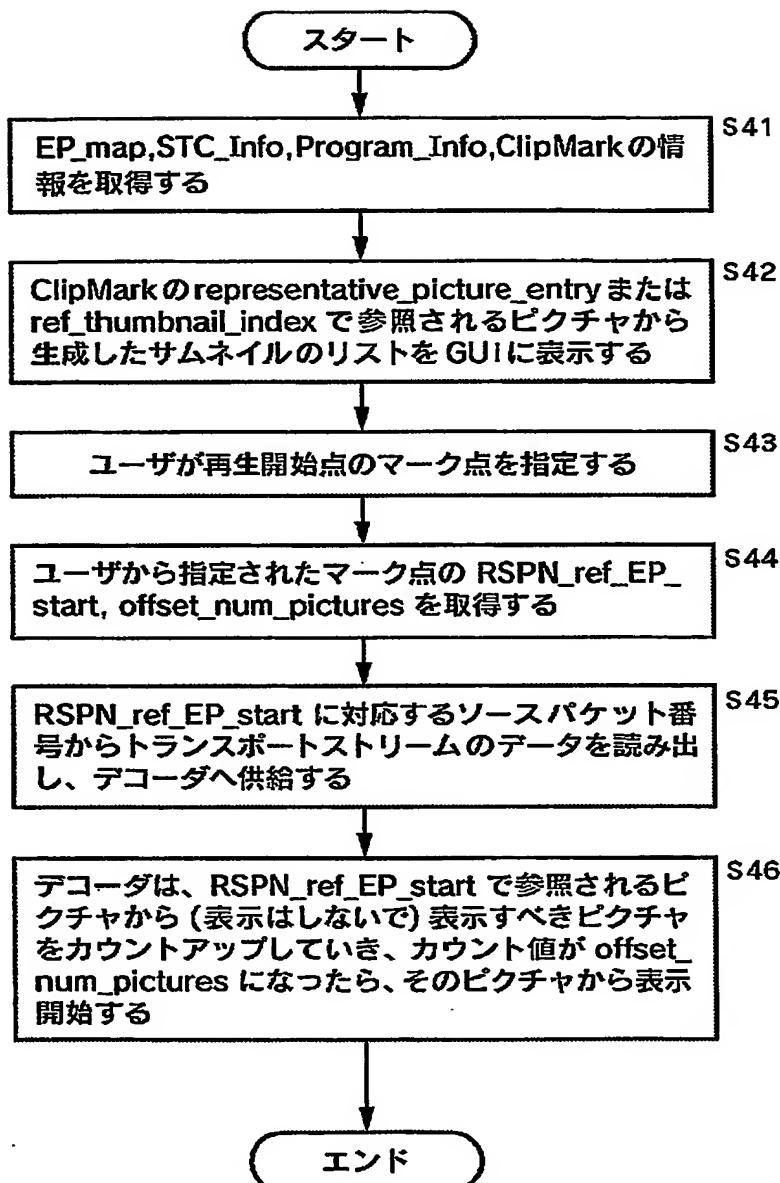
【図110】



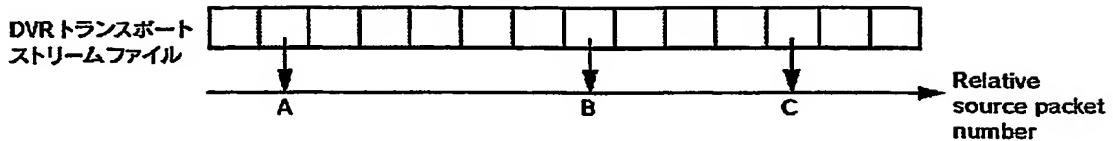
【図111】



【図112】



【図113】



【図114】

EP_map

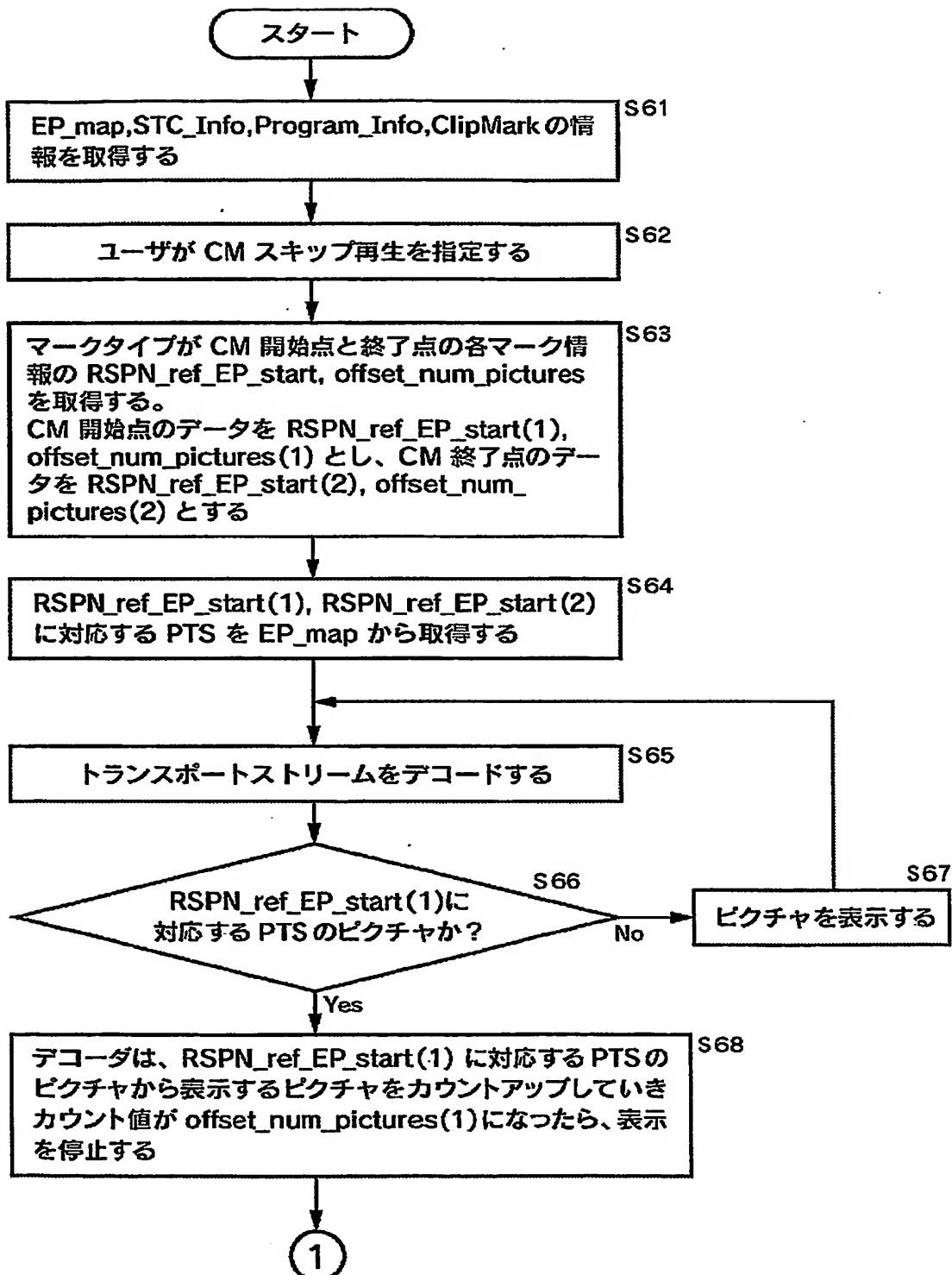
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

【図115】

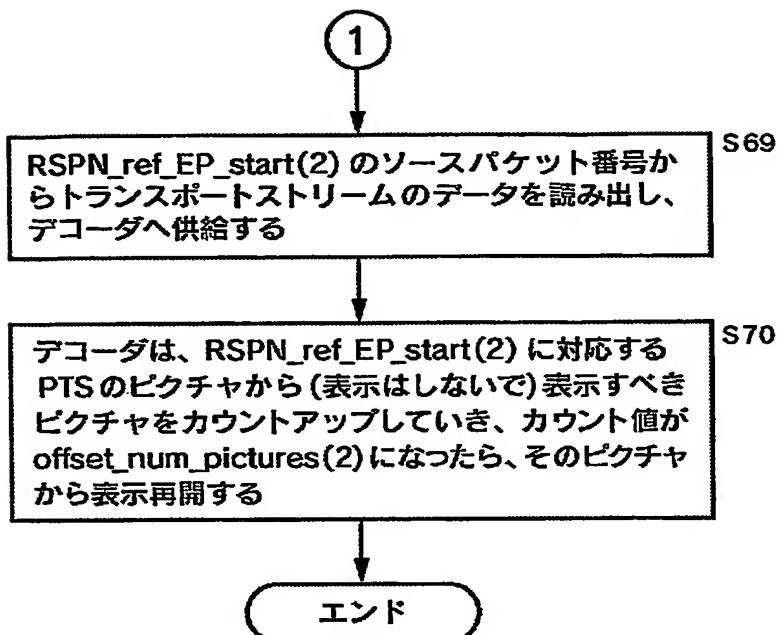
ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...

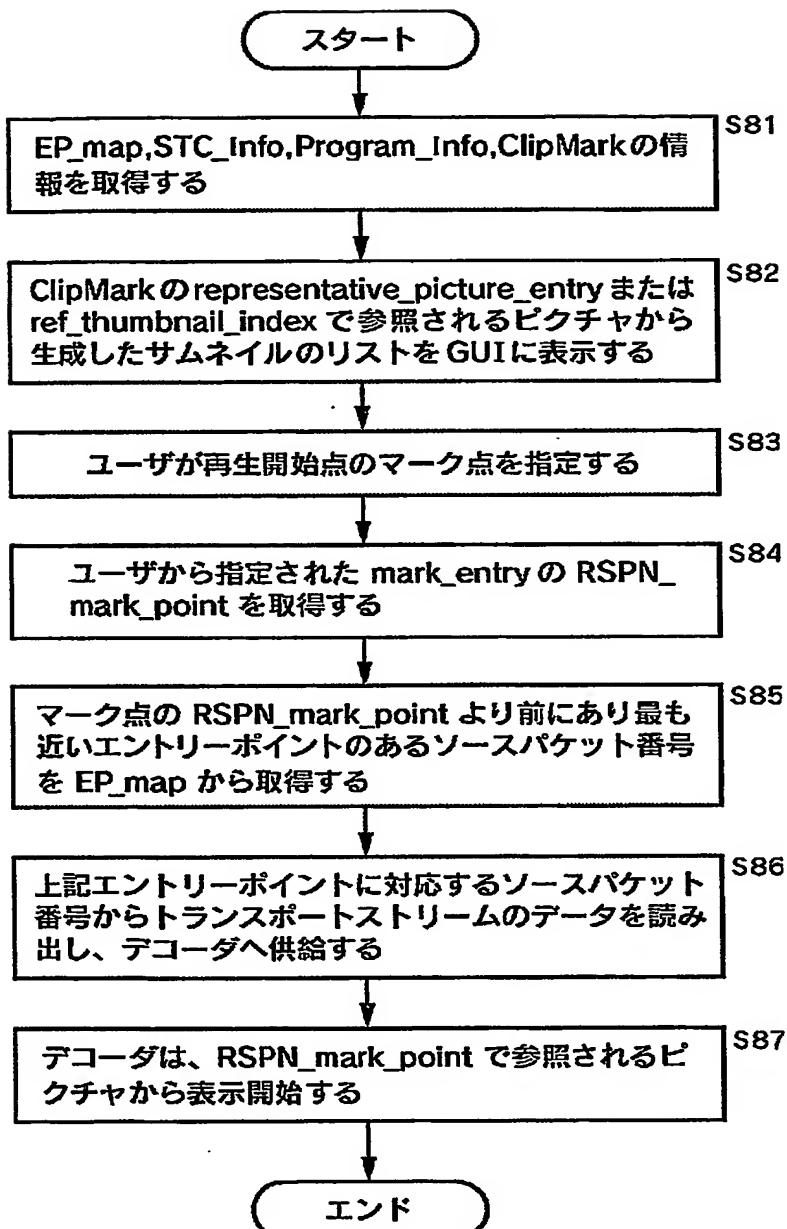
【図116】



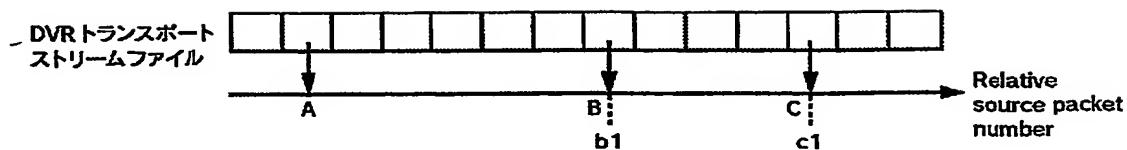
【図117】



【図118】



【図119】



【図120】

EP_map

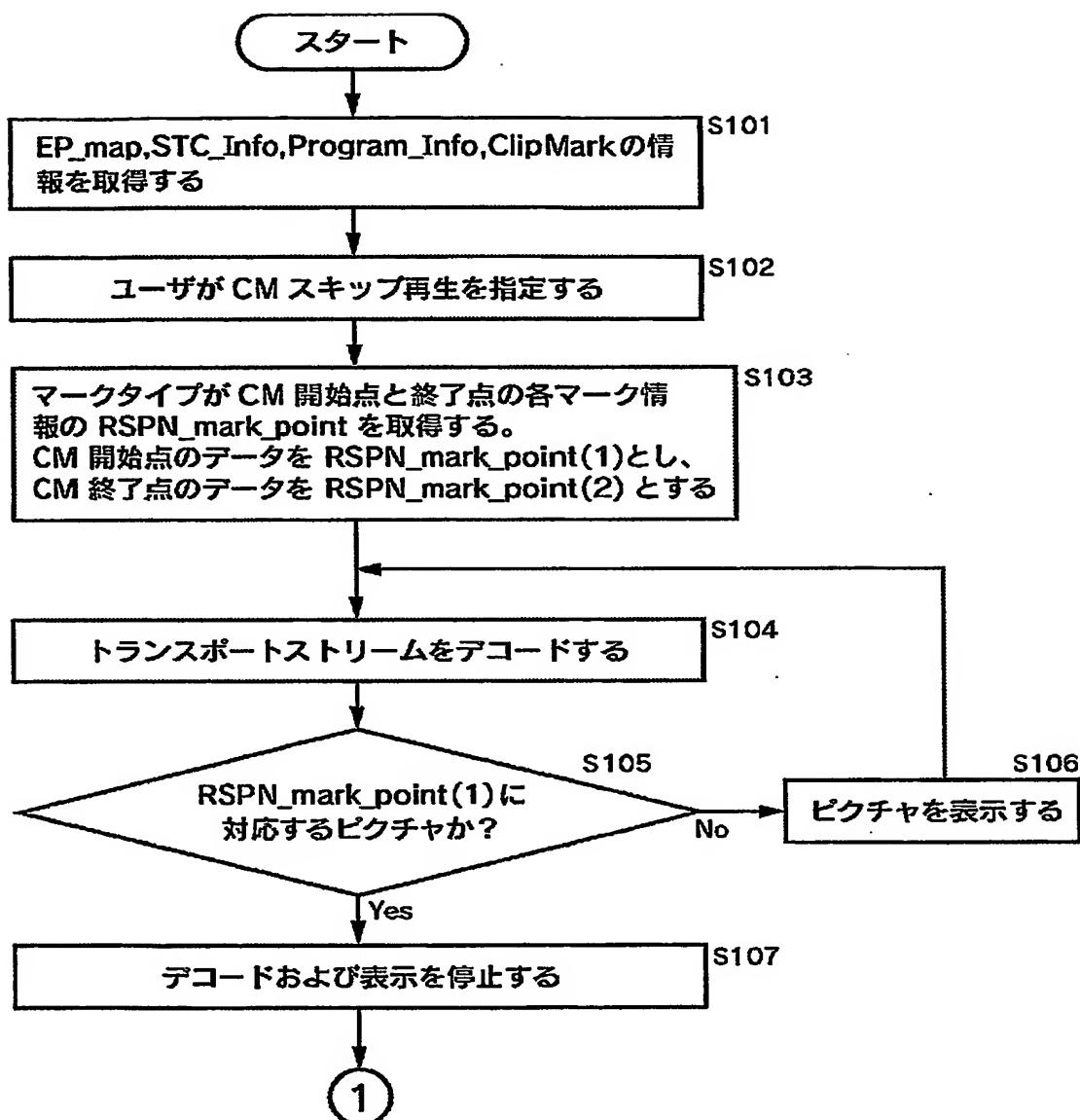
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

【図121】

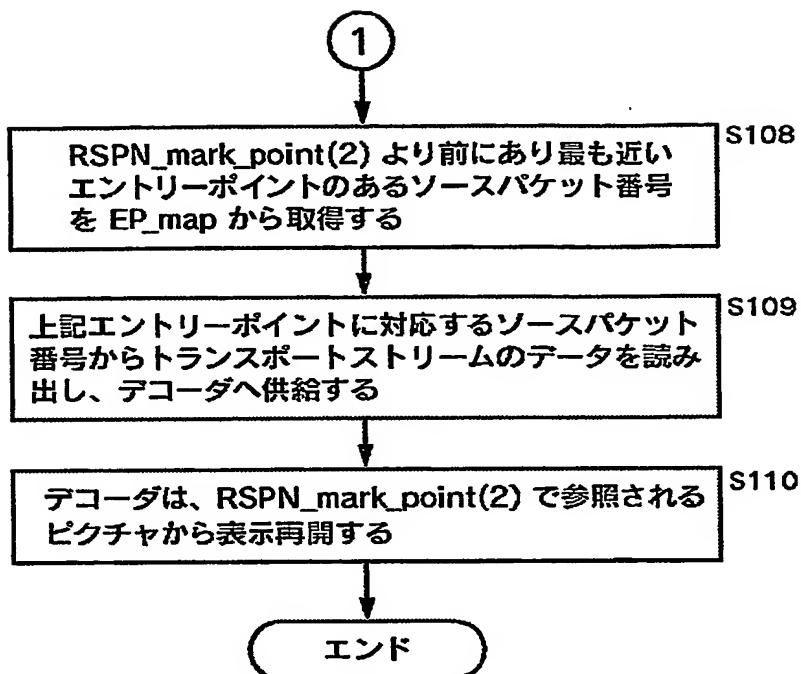
ClipMark

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...
0x92(scene start)	a1	a2
0x94(CM start)	b1	b1
0x95(CM end)	c1	c1
...

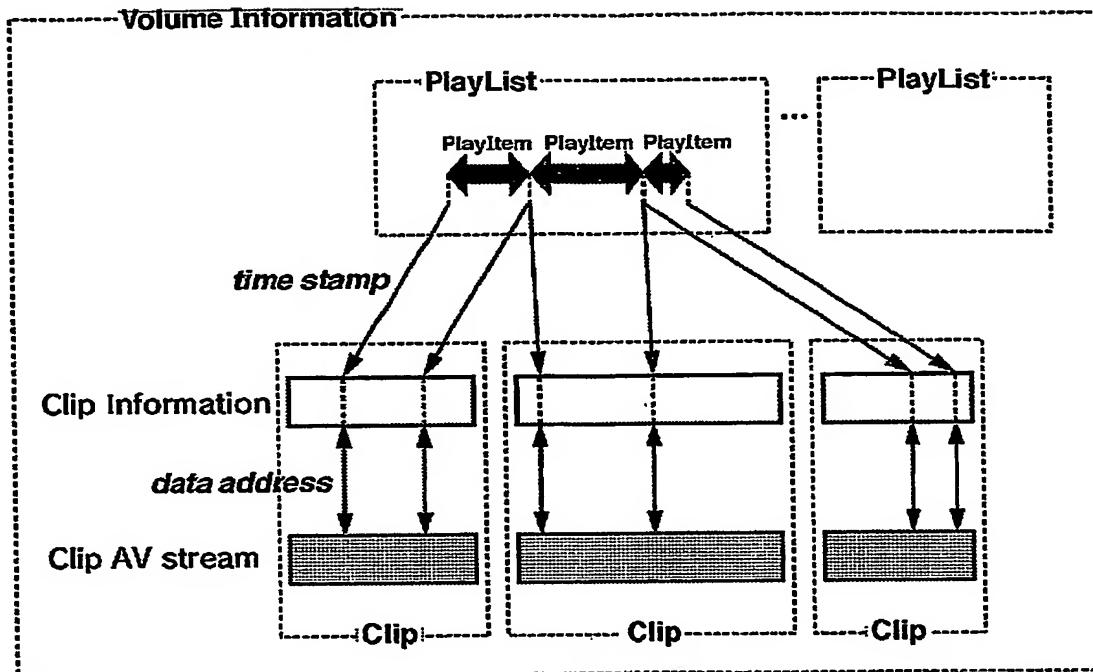
【図122】



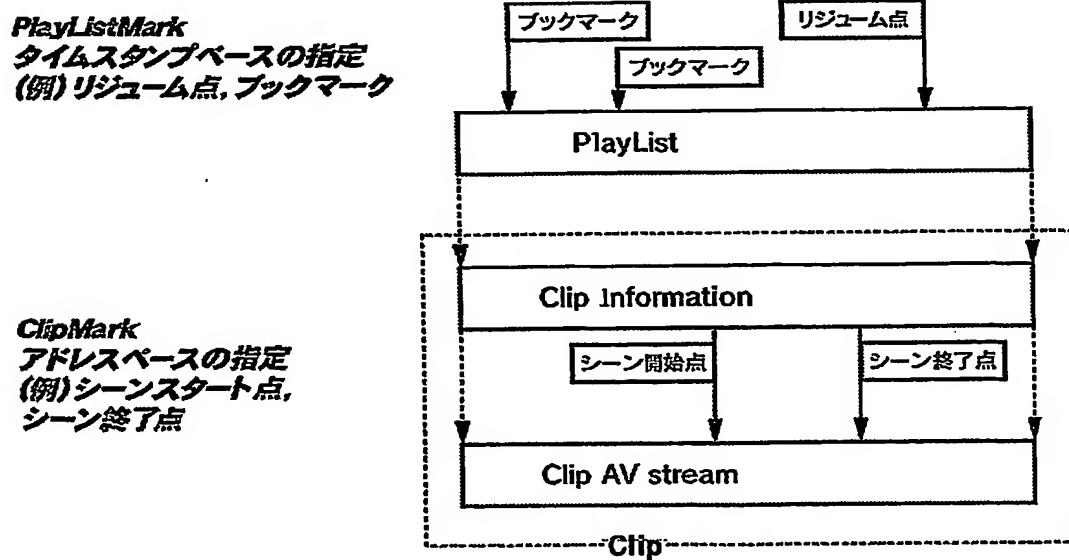
【図123】



【図124】



【図125】



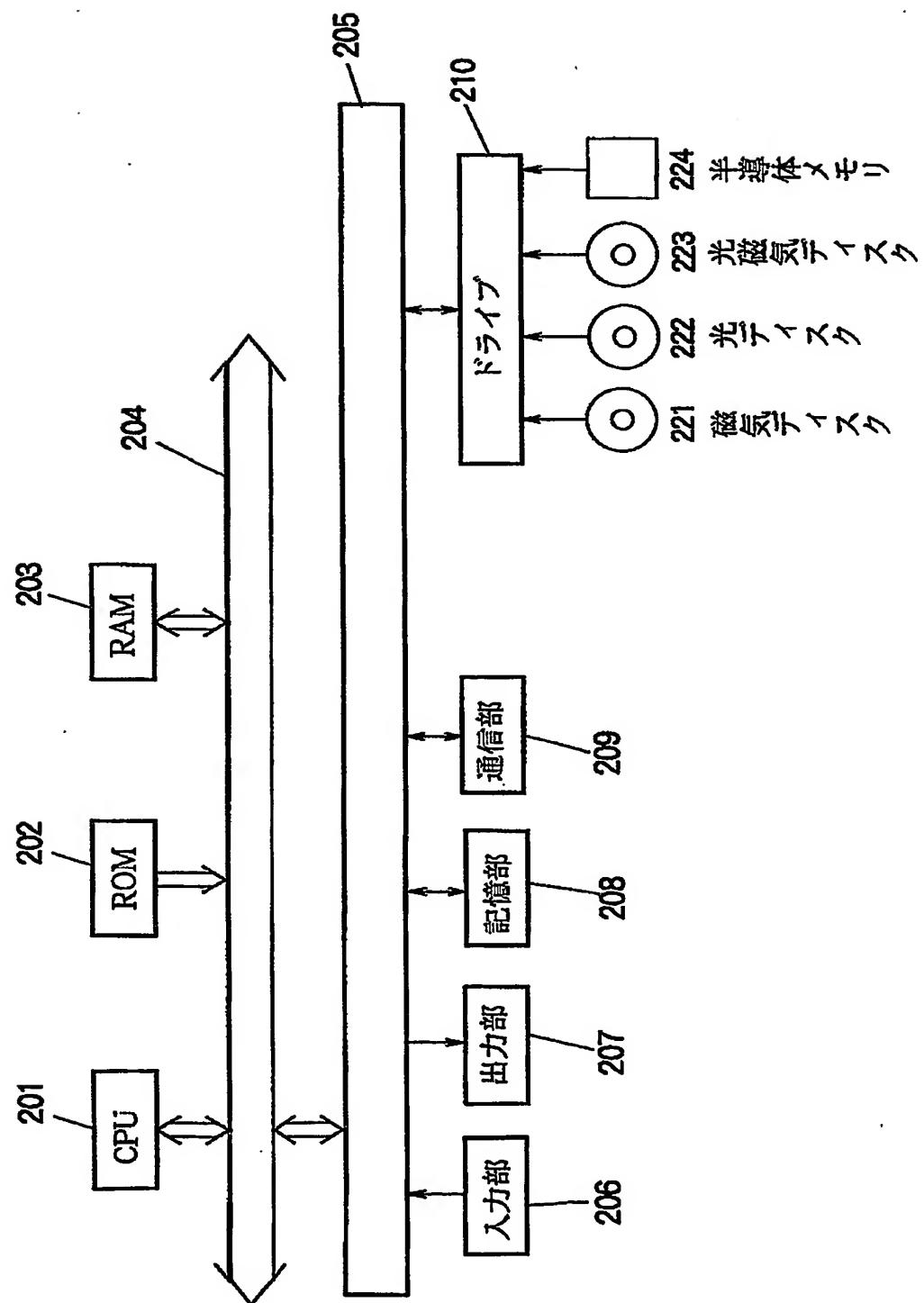
【図126】

Syntax	No. of bits	Mnemonics
ClipMark()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0;i<number_of_Clip_marks;i++) {		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_mark	32	uimsbf
reserved	32	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

【図127】

Syntax	No. of bits	Mnemonics
ClipMark()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0;i<number_of_Clip_marks;i++) {		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
ref_thumbnail_index	16	uimsbf
}		
}		

【図128】



【書類名】 要約書

【要約】

【課題】 AVストリームの所望の位置に迅速且つ確実にアクセスできるようにする。

【解決手段】 AVストリームを構成するClipのうち、プログラム（番組）の開始点は、`mark_entry()` に記述され、プログラムのタイトルが表示されているピクチャは、`representative_picture_entry()` に記述される。

【選択図】 図80

認定・付加情報

特許出願の番号	特願2000-268043
受付番号	50001129906
書類名	特許願
担当官	第七担当上席 0096
作成日	平成12年 9月 8日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000002185
【住所又は居所】	東京都品川区北品川6丁目7番35号
【氏名又は名称】	ソニー株式会社

【代理人】

【識別番号】	100082131
【住所又は居所】	東京都新宿区西新宿7丁目5番8号 GOWA西 新宿ビル6F 稲本国際特許事務所
【氏名又は名称】	稻本 義雄

次頁無

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社